TOWARD A NEW PROOF OF THE FOUR COLORING THEOREM

JOHN CLAIRMONT

Abstract. A heuristic algorithm is described for four coloring planar graphs. The algorithm is based on the principle of breaking a problem into components and solving them one at a time. The algorithm uses generalized alternating chains of the most general kind, for instance they can have branches and self intersections. Alternating chains are found which, after completing each major iteration of the algorithm, a certain property is preserved. This property ensures that a successive alternating chain will be easy to find.

Specifically the property ensures that the branches of an alternating chain cannot conflict with each other, and a branch cannot conflict with itself. This algorithm solves one set of problems, but the solution has it's own set of problems. However I think these can be easily resolved, leading to a new and simple proof of the Four Color Theorem.

1. An example

In the beginning a checker of each of the four colors is placed on every vertex. Things start off slowly, the first alternating chain of length greater than one occurs at figure 64. Property B, the crux of the algorithm, is defined at figure 4.
(Step through the example page by page so you can see the algorithm in action - don't scroll.)

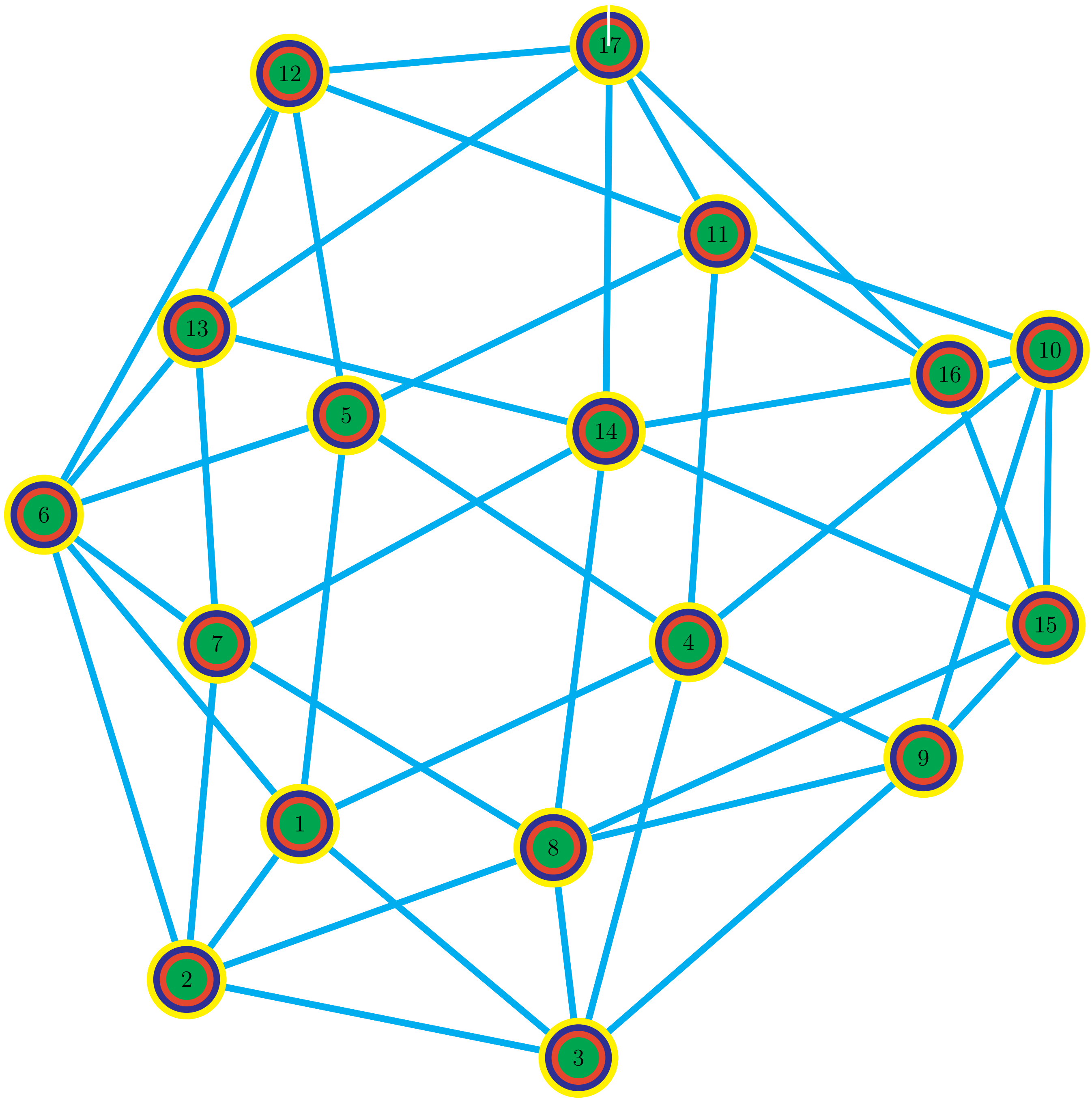

Figure 1. Instructions:

In the beginning a checker of each of the four colors, Green, Red, Blue, and Yellow, is placed on every vertex.

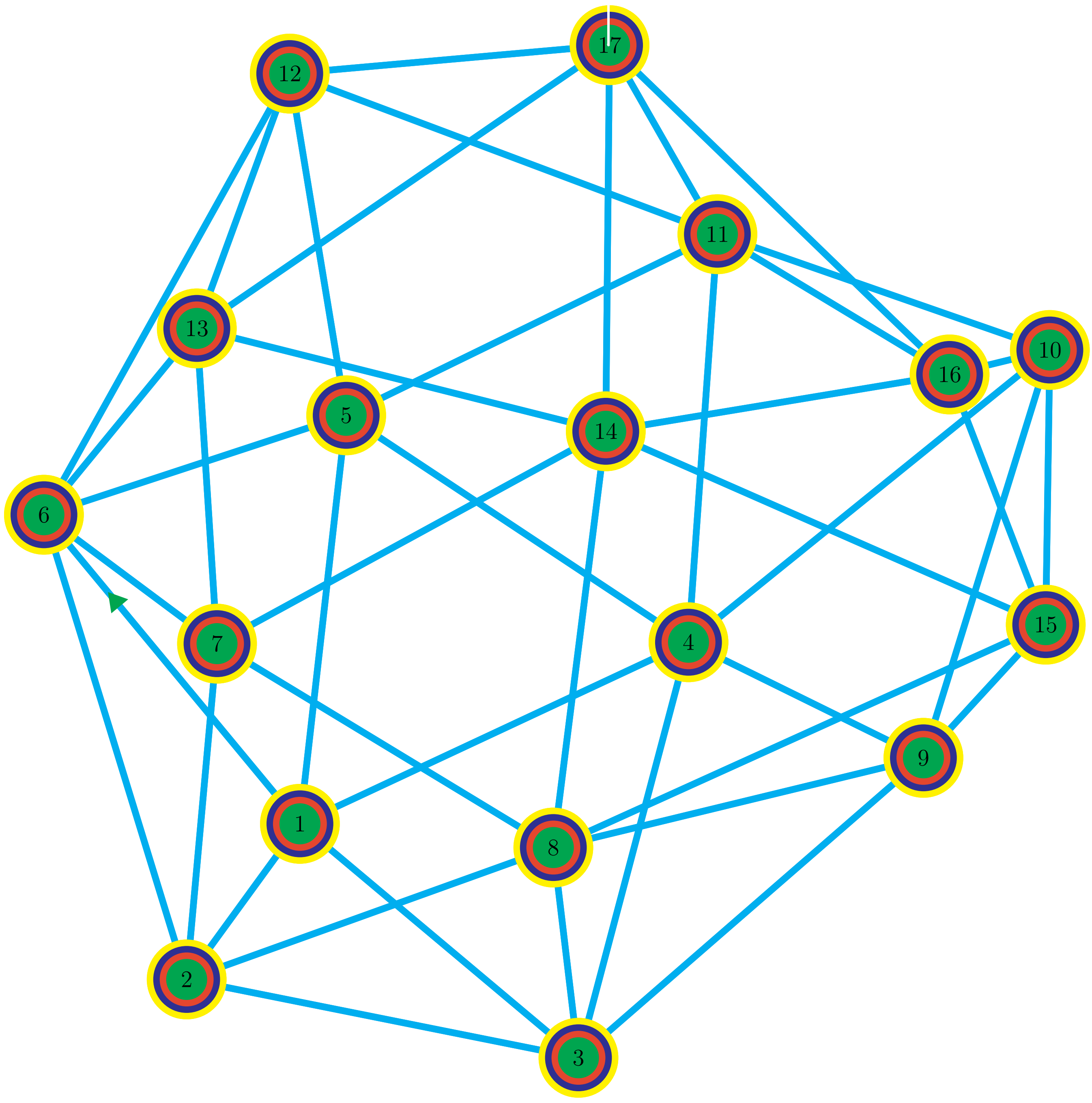

Figure 2. Instructions: 1: place edge 1→6 Green ArrowR,

When the above instruction is executed by the computer a triangular green arrowhead is placed on the directed edge 1→6. A triangular green arrowhead is used to indicate the deletion(Removal) of the green checker on the vertex pointed to (in this case vertex 6). Define triangle arrowheads as deletion(Removal) arrows. Later we'll define insertion arrows.

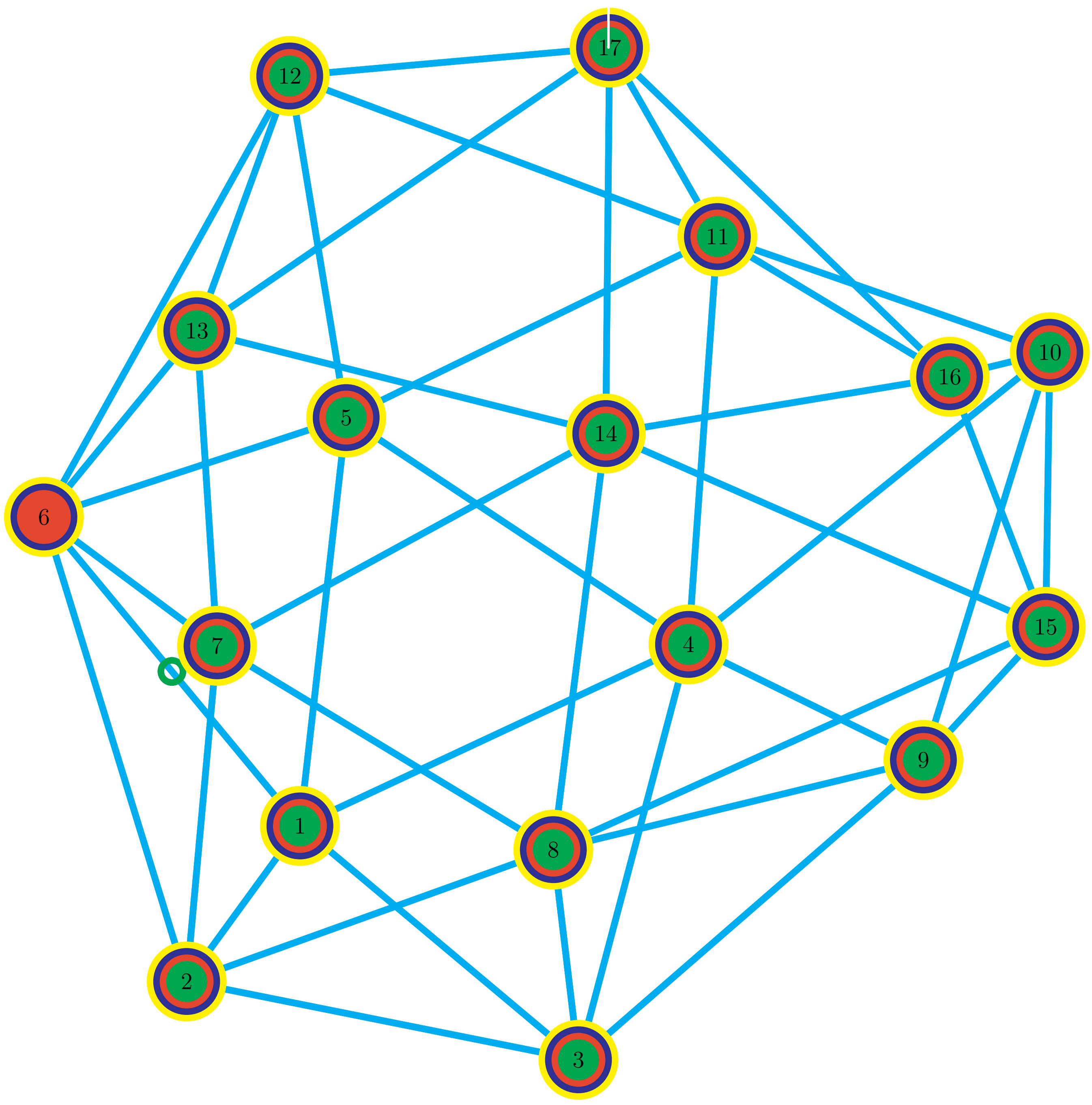

Figure 3. Instructions: 4: color edge 1–6 Green, 5: uncolor vertex 6 Green,

When the above instructions are executed a circle is placed(exactly in the middle) on the undirected edge 1-6 indicating that this edge is green colored. The green checker has been removed from vertex 6. Now the edge is conflict free wtr the color green ie there is only one green checker on vertices of this edge.

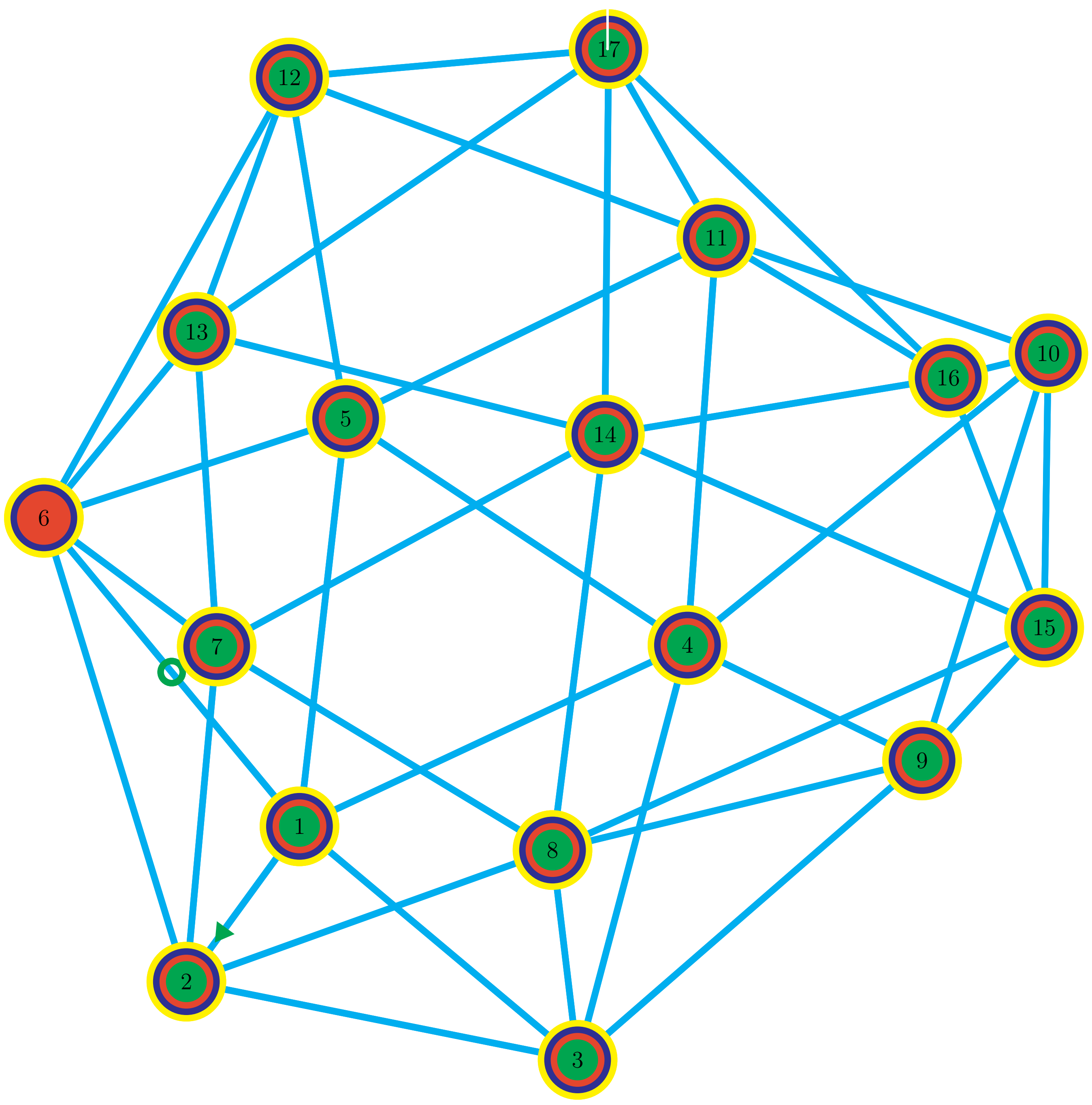

Figure 4. Instructions: 7: place edge 1→2 Green ArrowR,

Continuing in this manner we obtain a sequence of graphs with the following properties.

1: If a circle of a given color is on an edge there cannot be two checkers of the same color on vertices of that edge. Lets call this property A.

2: If a circle of a given color is on an edge there must be one and only one checker of that color on verices of that edge. Lets call this property B.

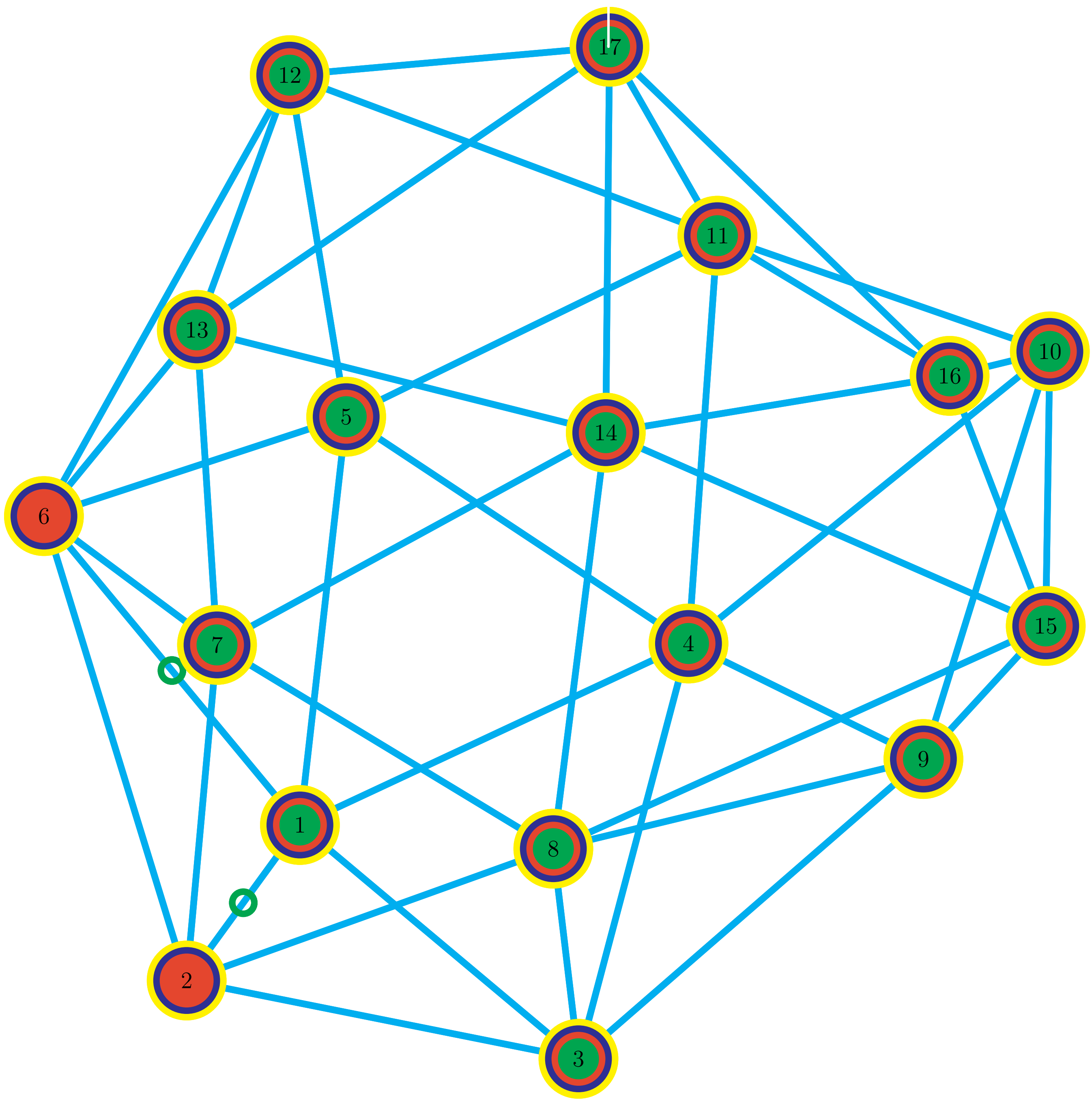

Figure 5. Instructions: 10: color edge 1–2 Green, 11: uncolor vertex 2 Green,

The green checker has been removed from vertex 2 and a green circle has been placed on edge 1-2 and properties A and B are are satisfied as they will be after each major iteration of the algorithm.

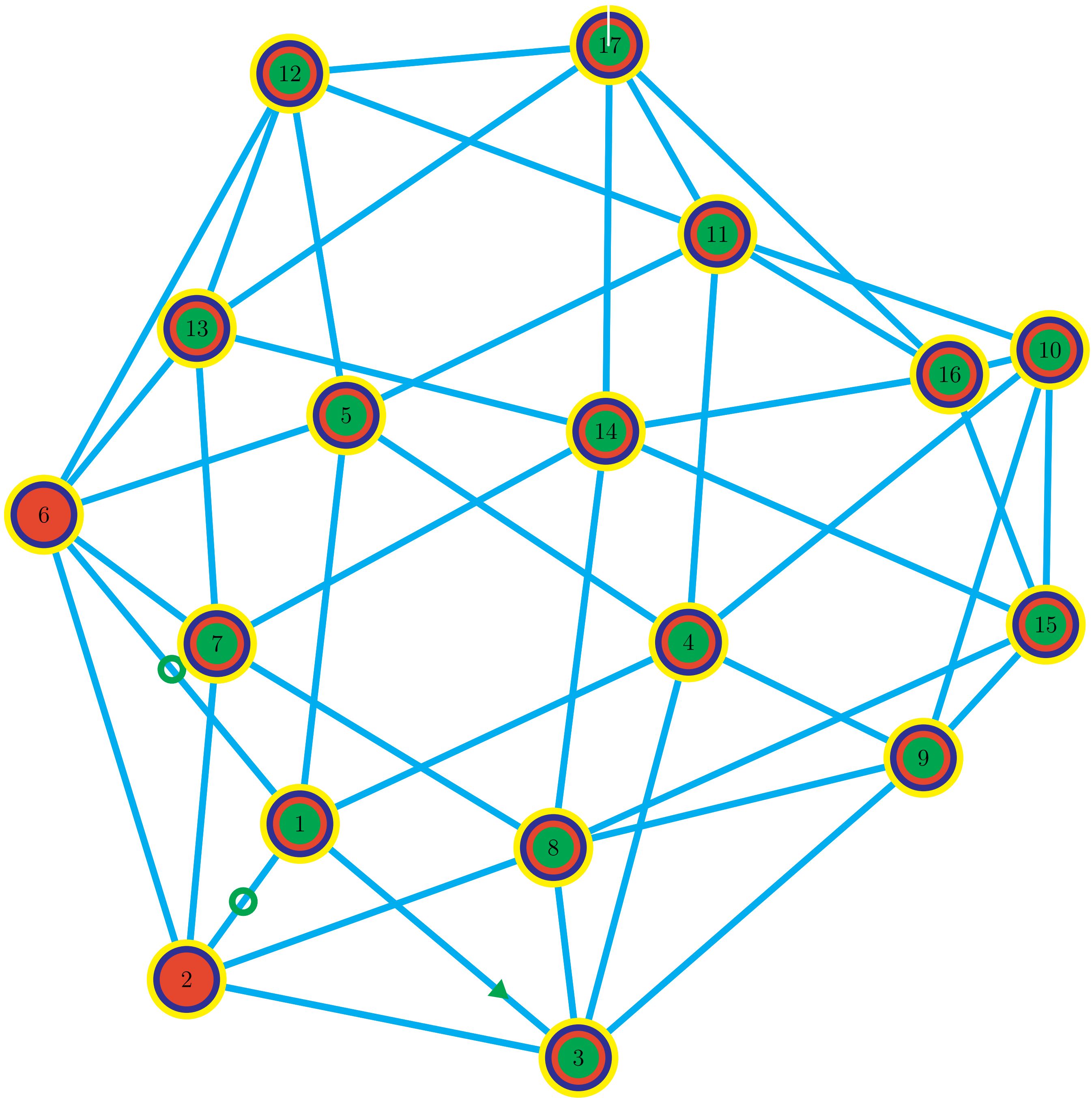

Figure 6. Instructions: 13: place edge 1→3 Green ArrowR,

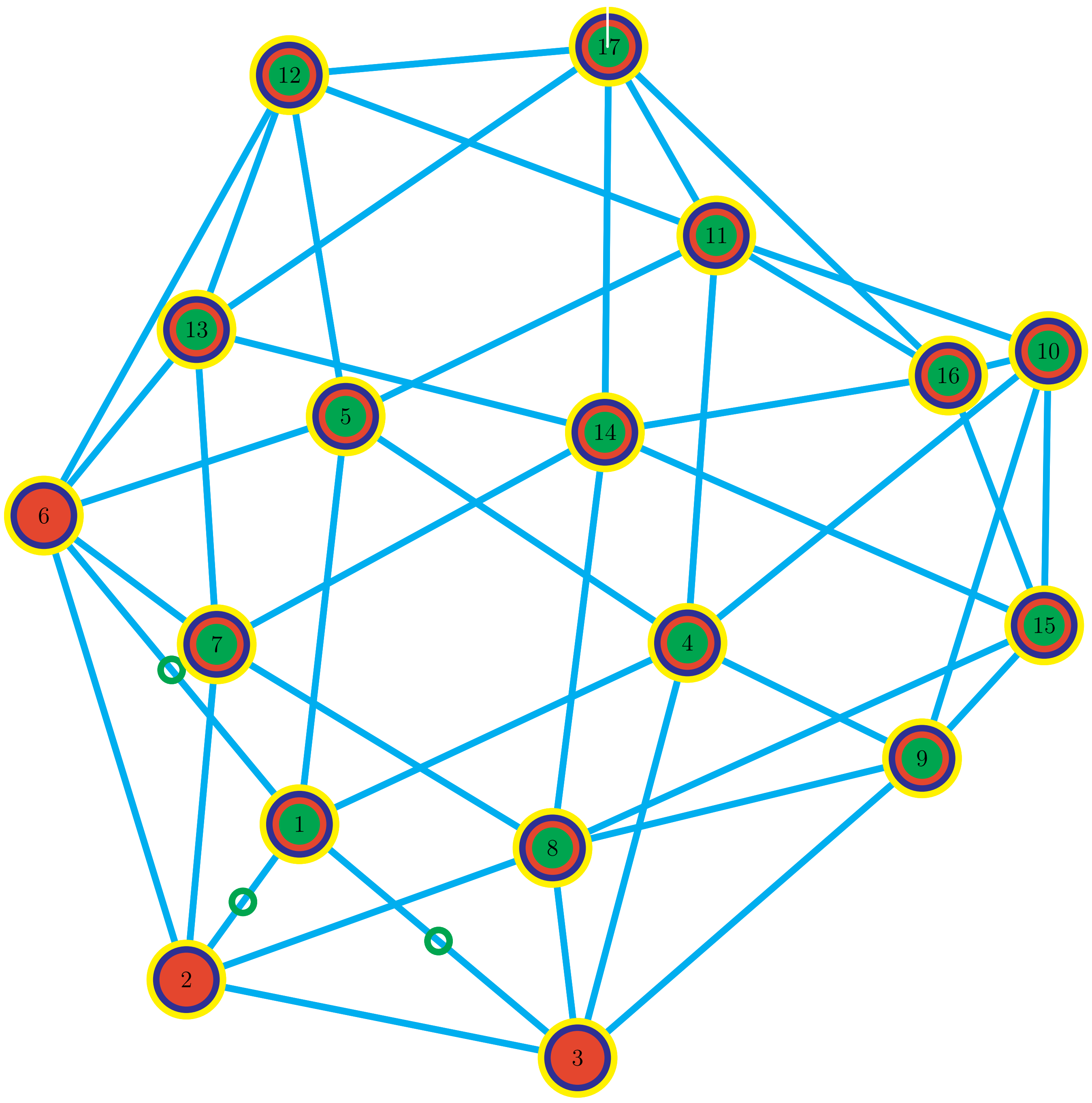

Figure 7. Instructions: 16: color edge 1–3 Green, 17: uncolor vertex 3 Green,

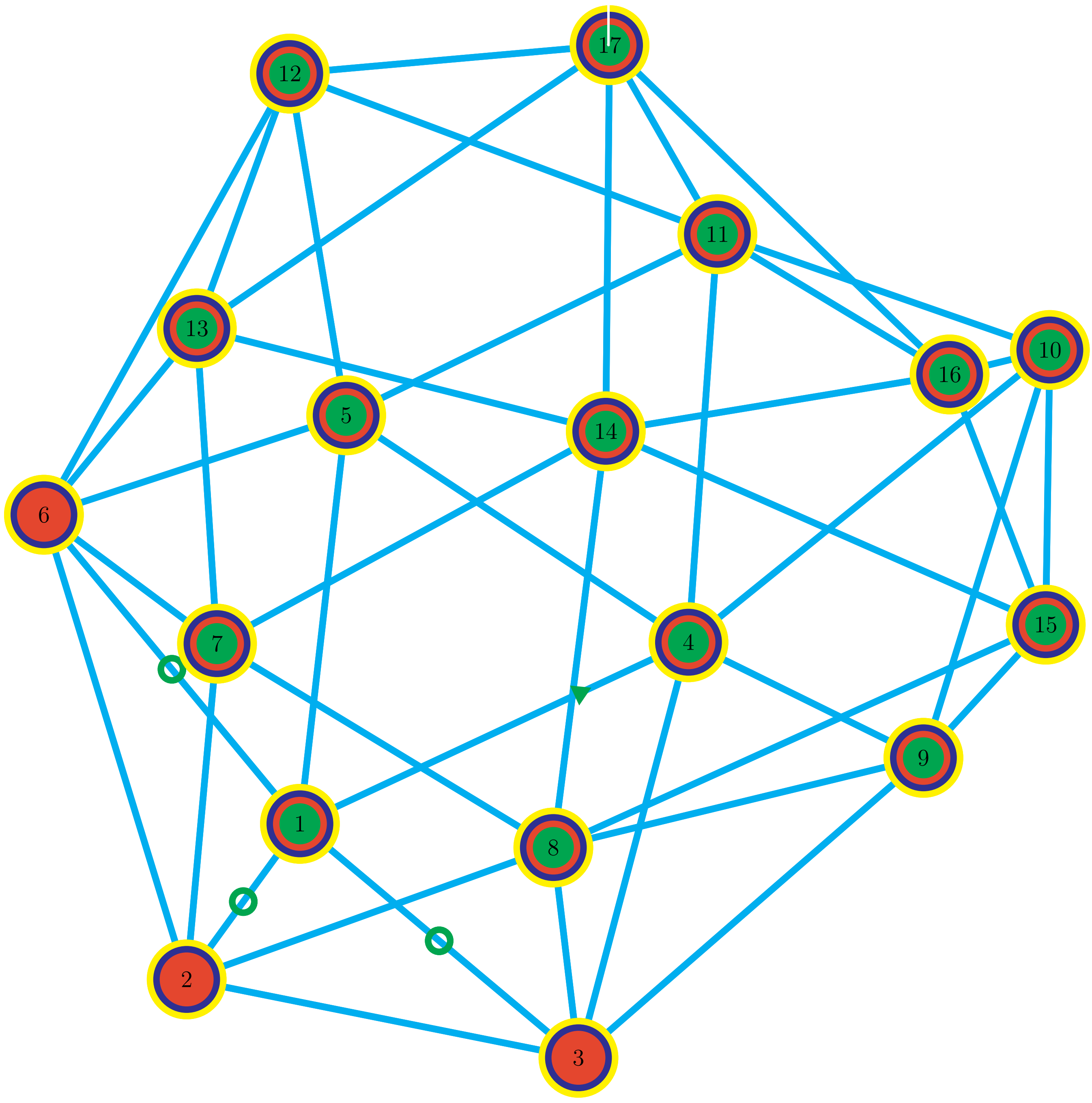

Figure 8. Instructions: 19: place edge 1→4 Green ArrowR,

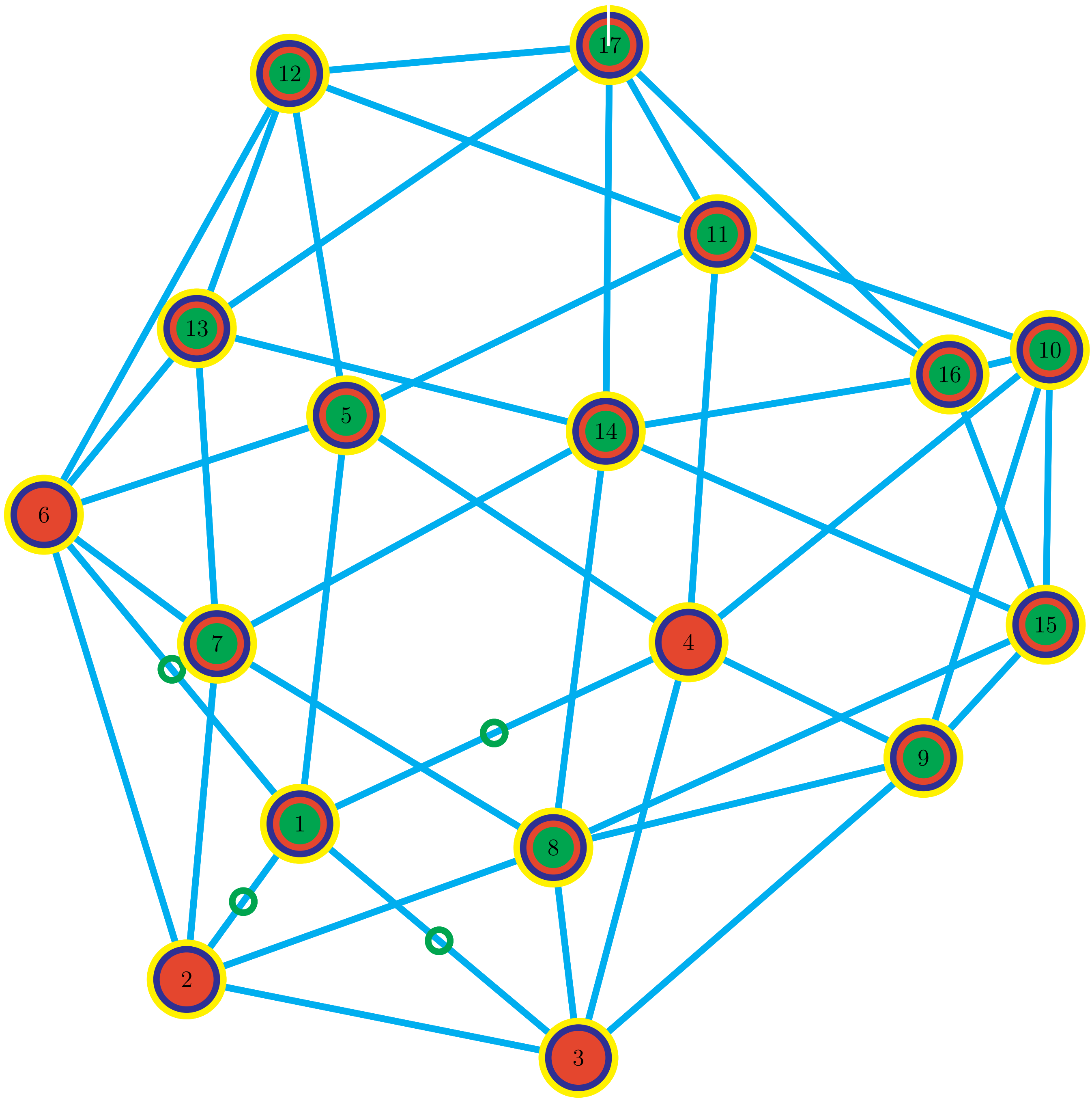

Figure 9. Instructions: 22: color edge 1–4 Green, 23: uncolor vertex 4 Green,

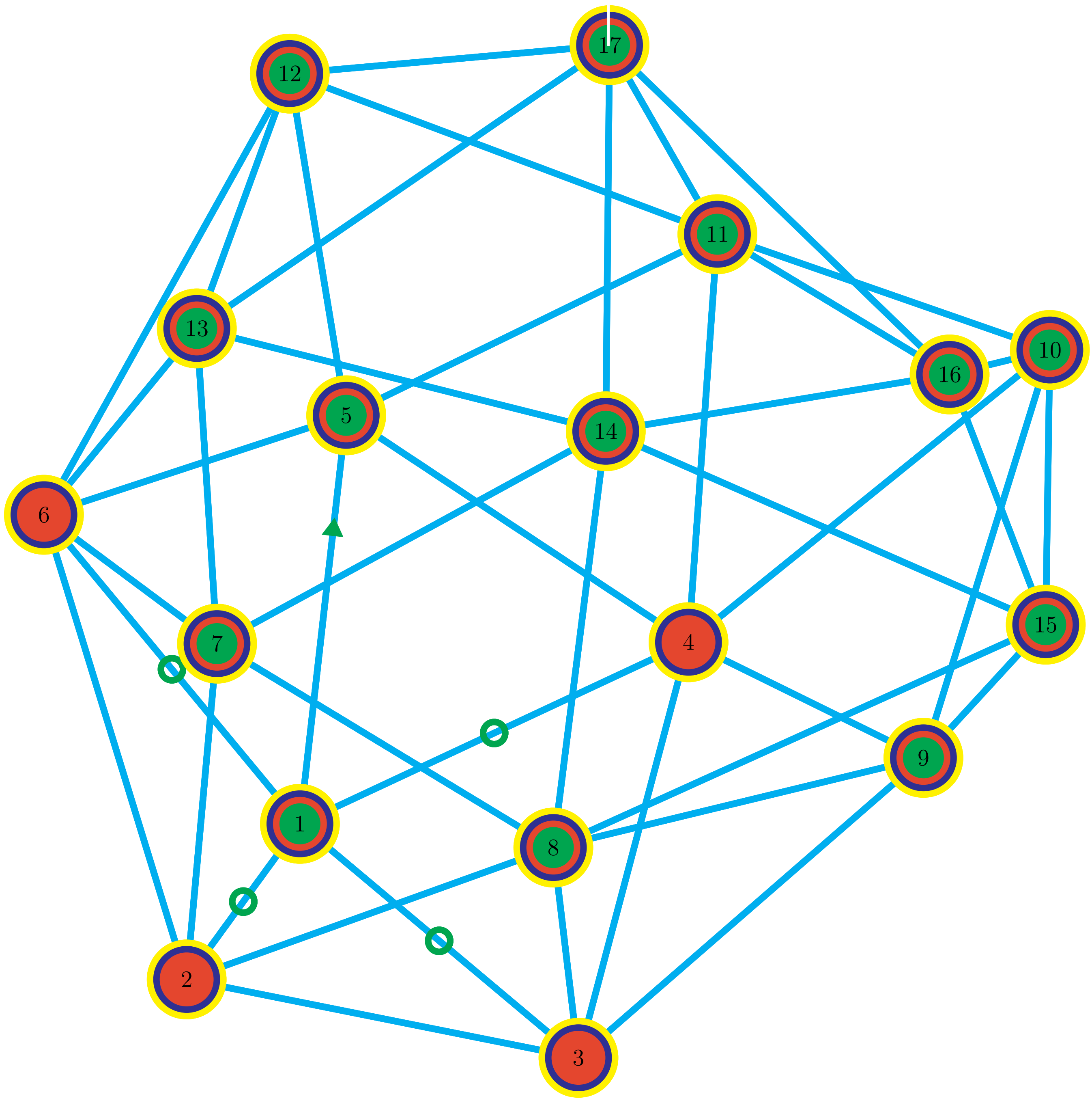

Figure 10. Instructions: 25: place edge 1→5 Green ArrowR,

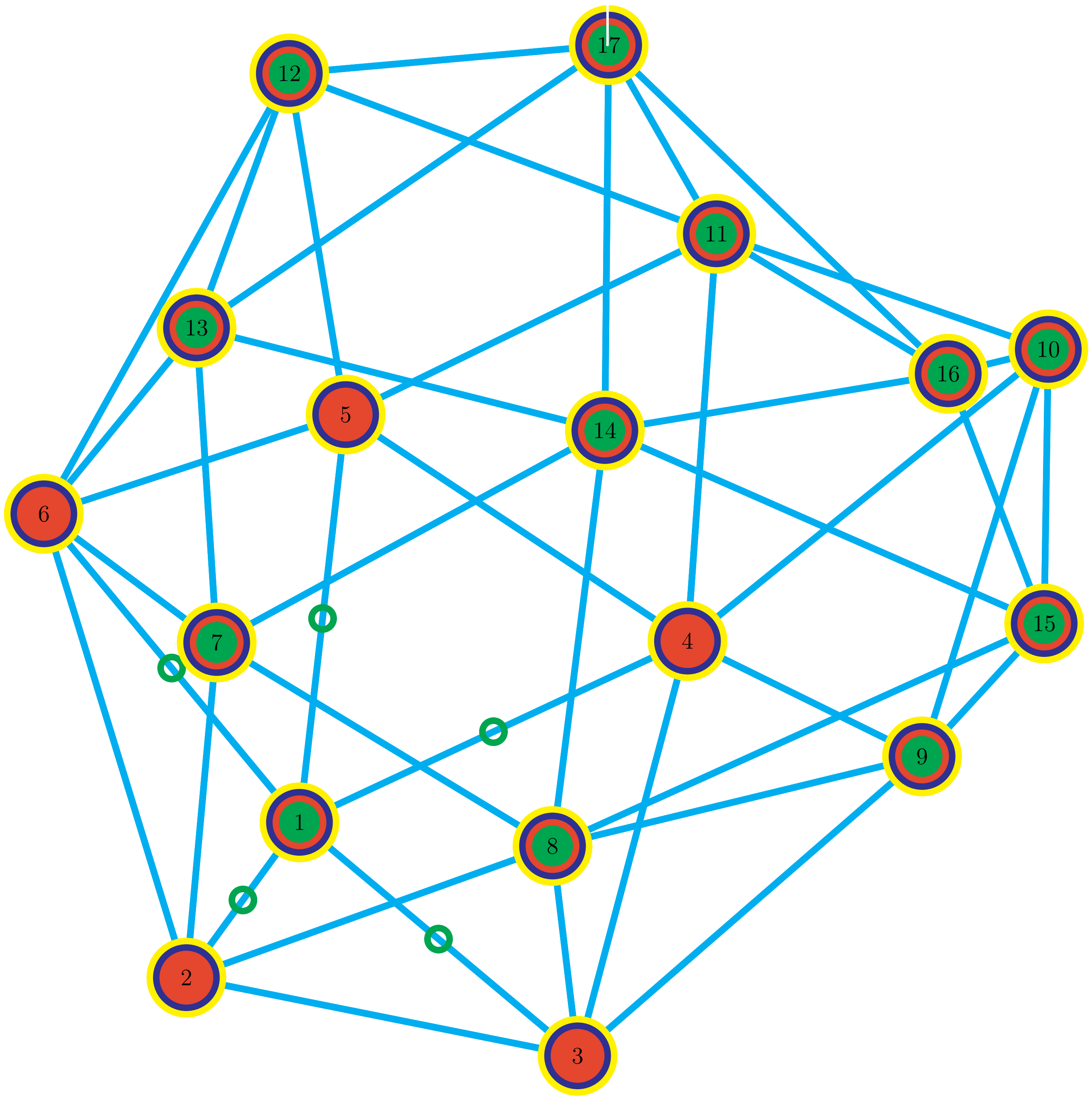

Figure 11. Instructions: 28: color edge 1–5 Green, 29: uncolor vertex 5 Green,

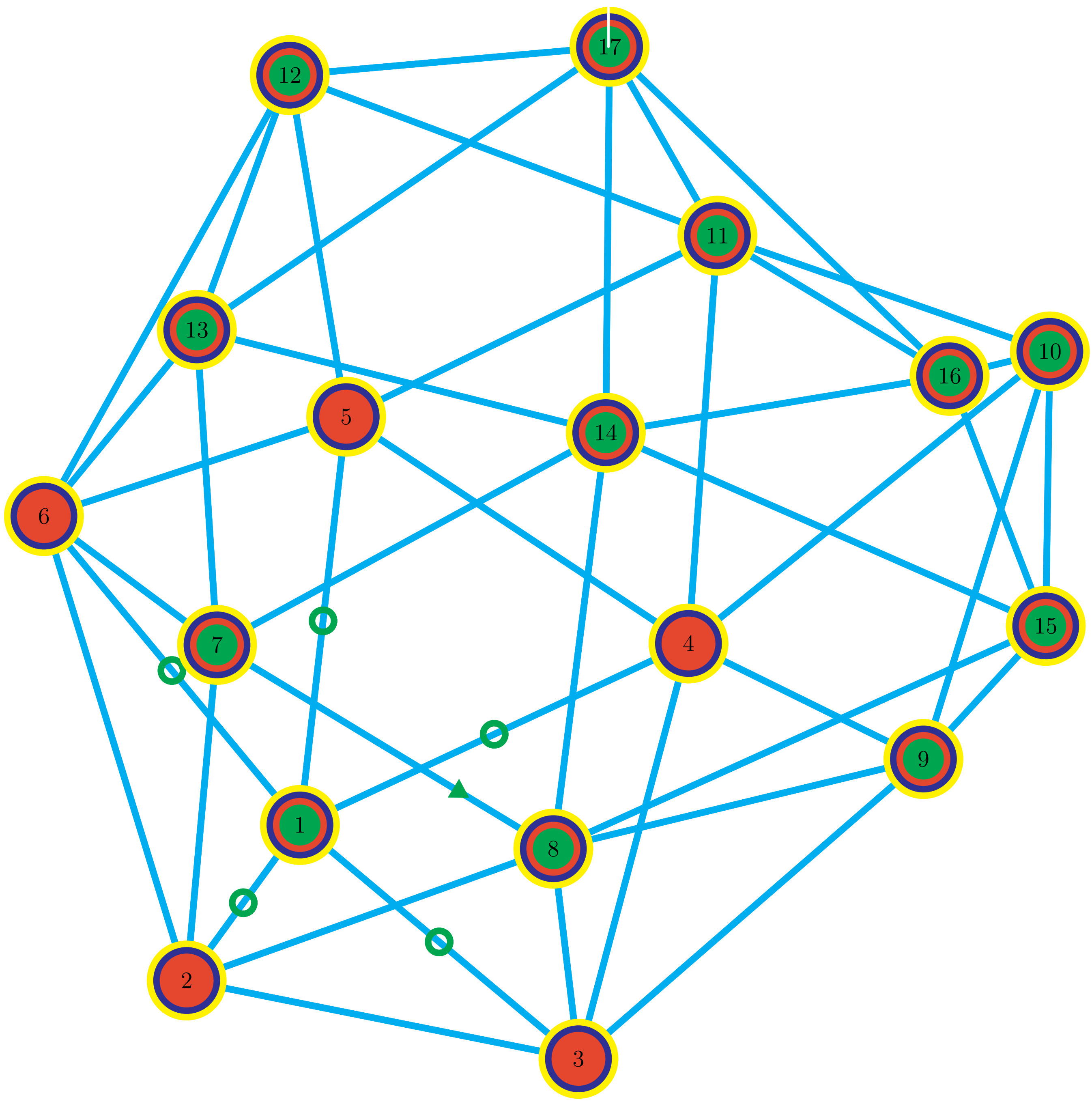

Figure 12. Instructions: 31: place edge 7→8 Green ArrowR,

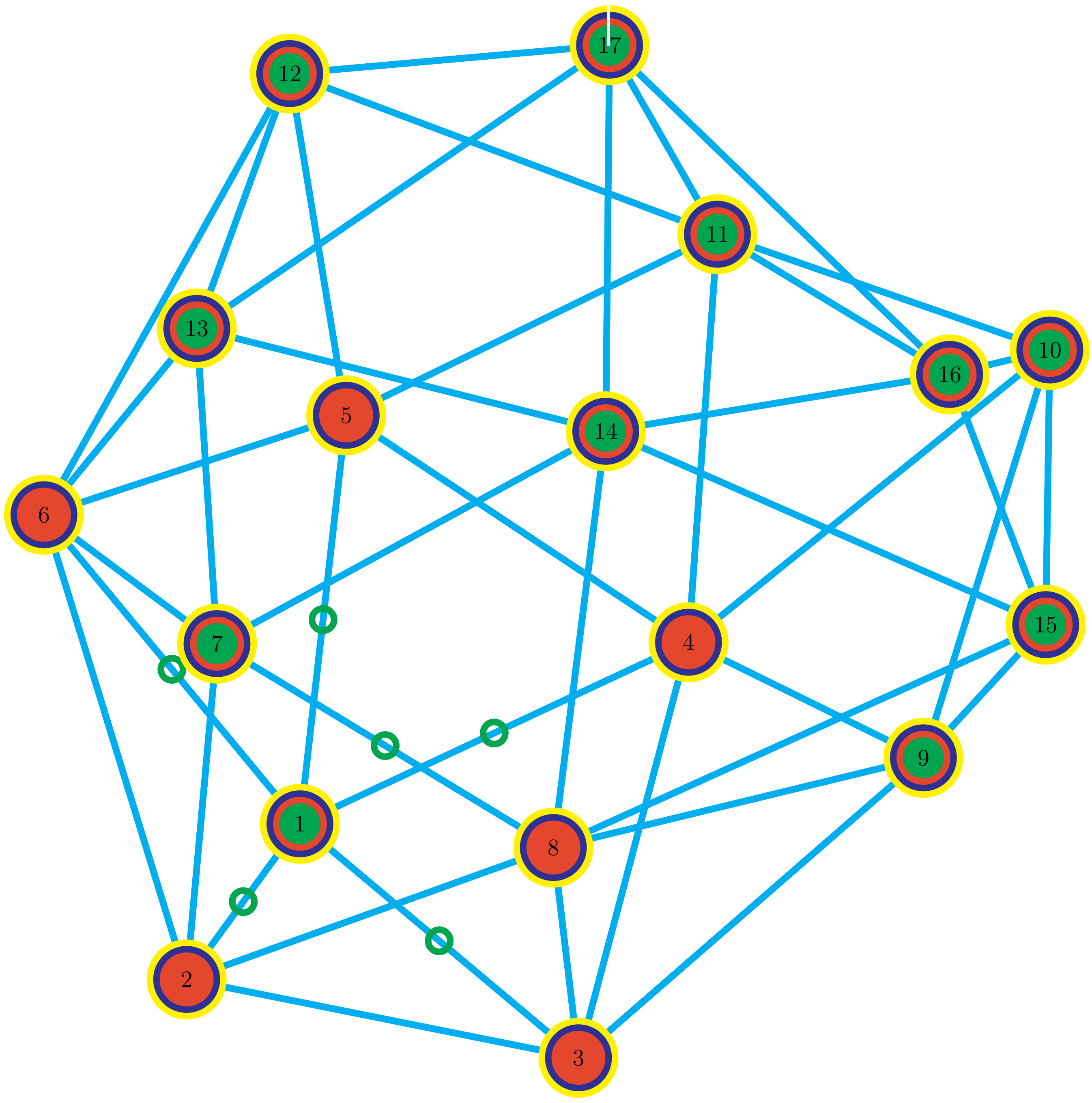

Figure 13. Instructions: 34: color edge 7–8 Green, 35: uncolor vertex 8 Green,

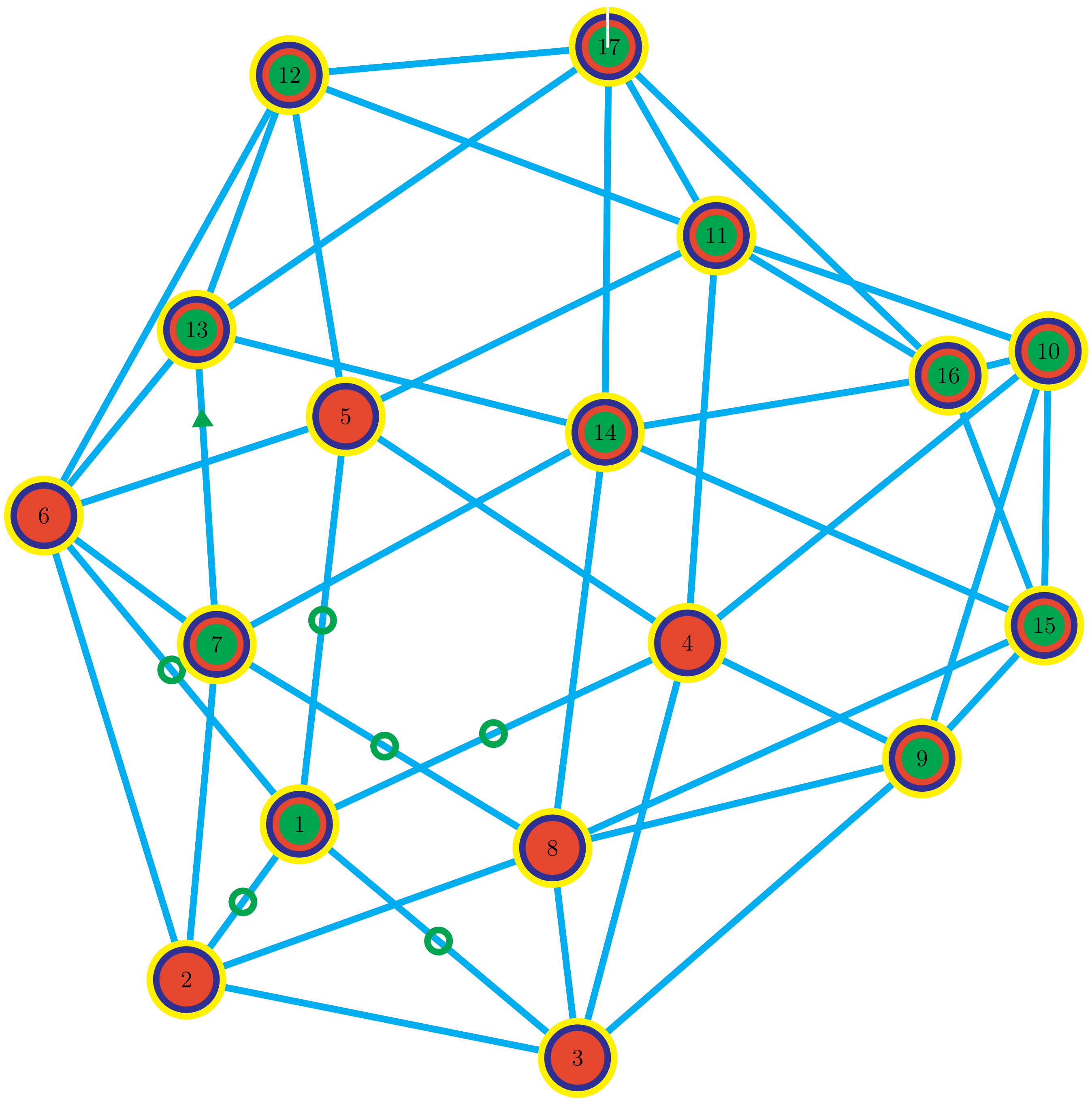

Figure 14. Instructions: 37: place edge 7→13 Green ArrowR,

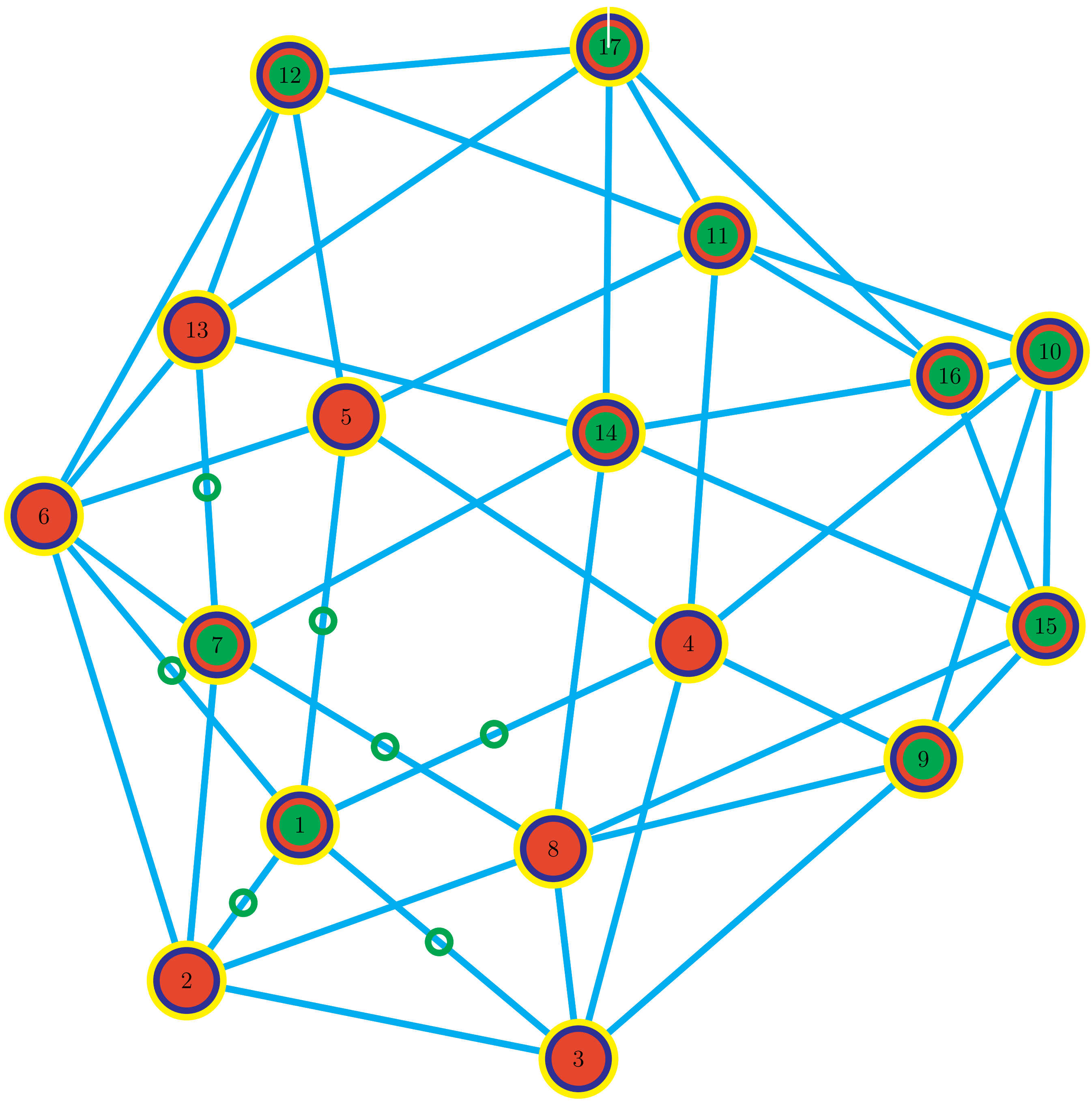

Figure 15. Instructions: 40: color edge 7–13 Green, 41: uncolor vertex 13 Green,

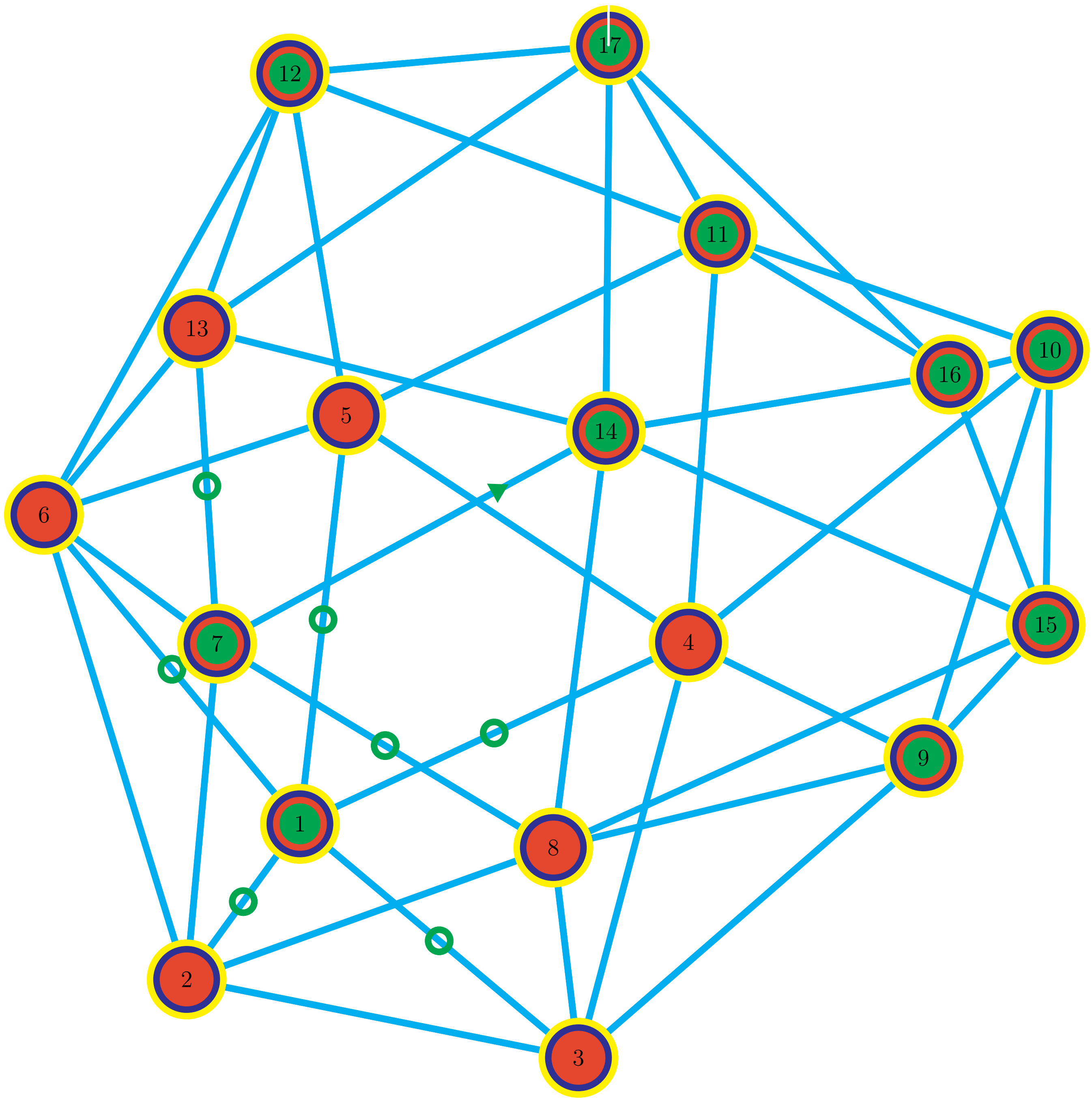

Figure 16. Instructions: 43: place edge 7→14 Green ArrowR,

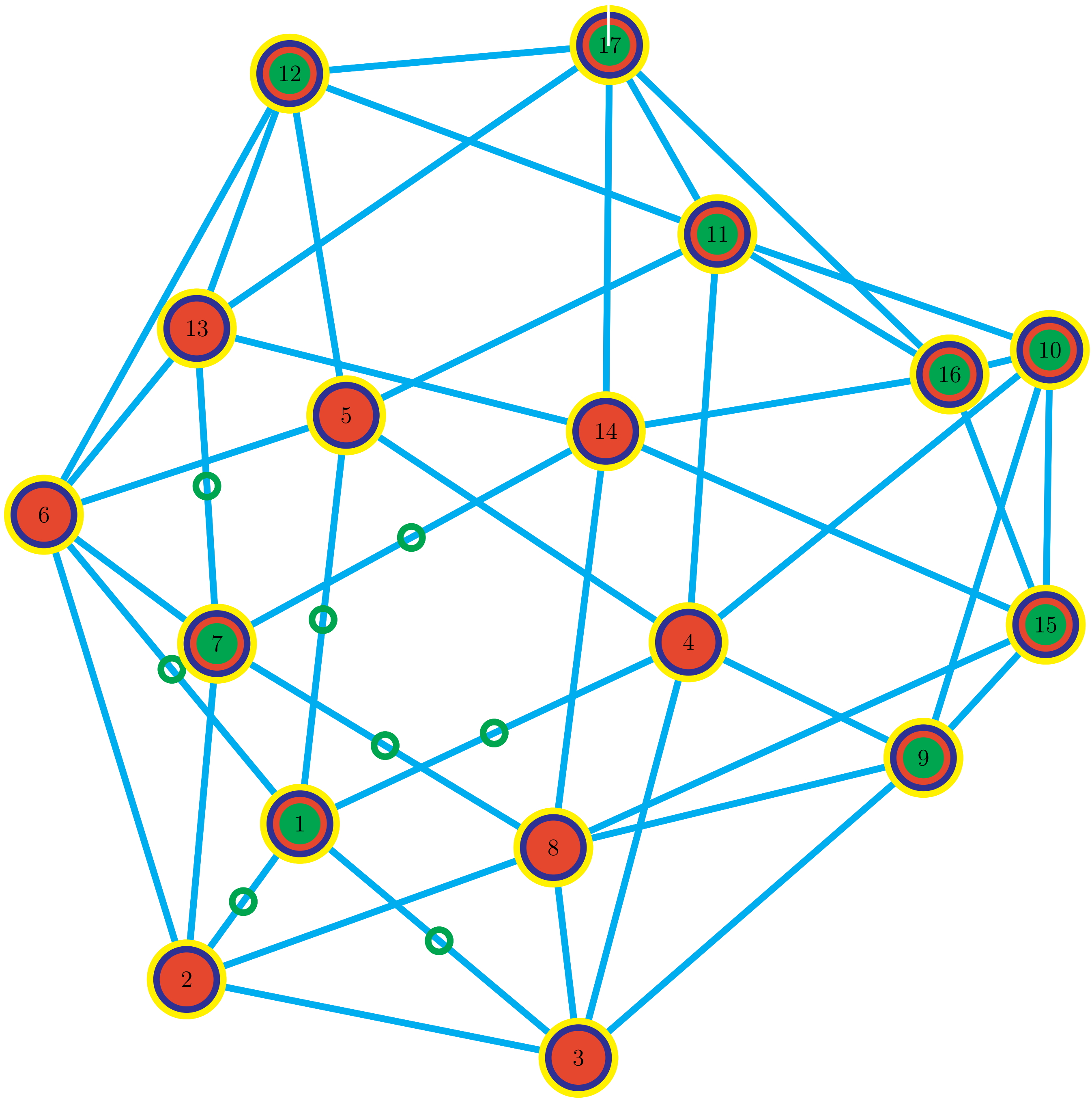

Figure 17. Instructions: 46: color edge 7–14 Green, 47: uncolor vertex 14 Green,

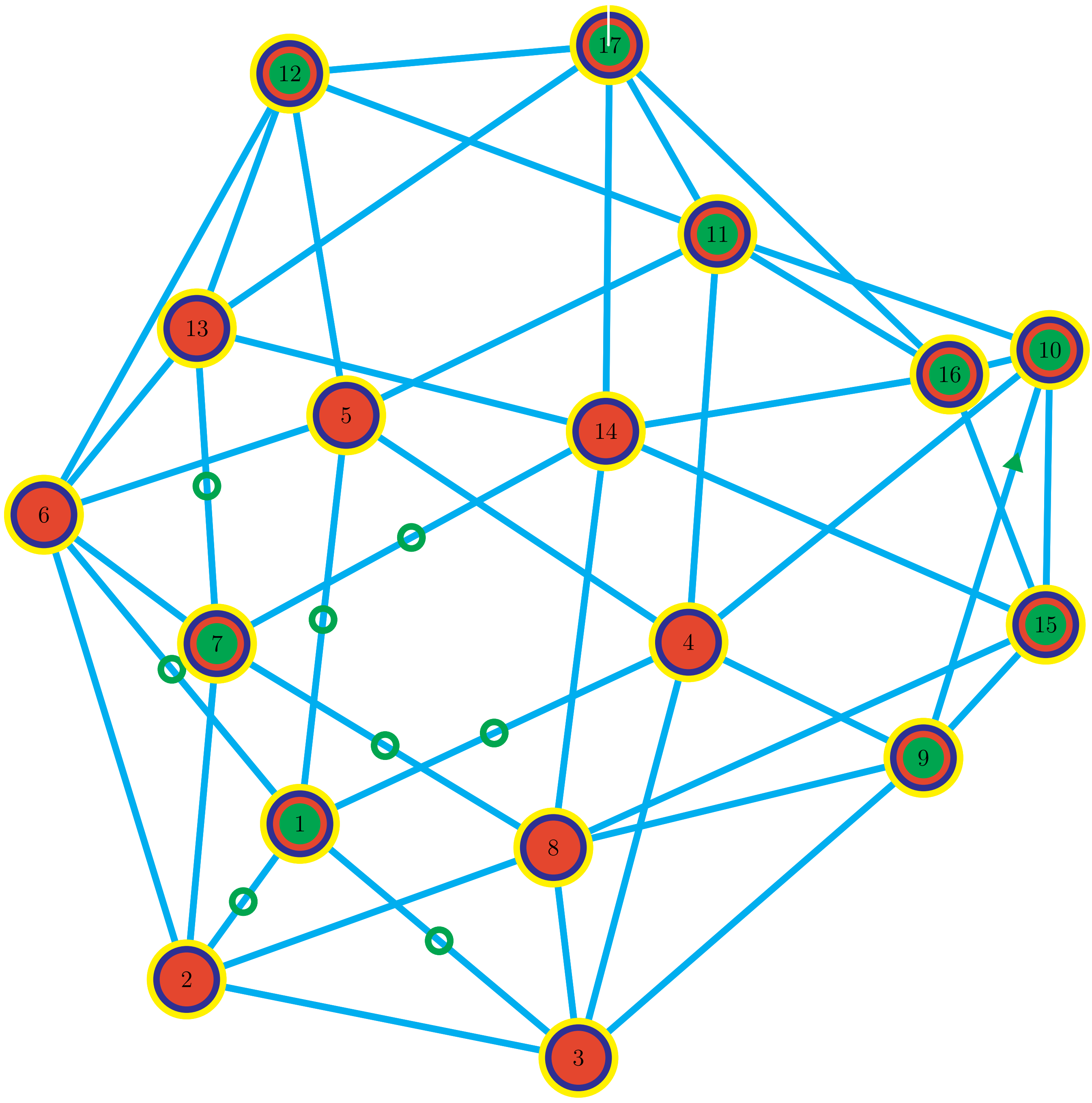

Figure 18. Instructions: 49: place edge 9→10 Green ArrowR,

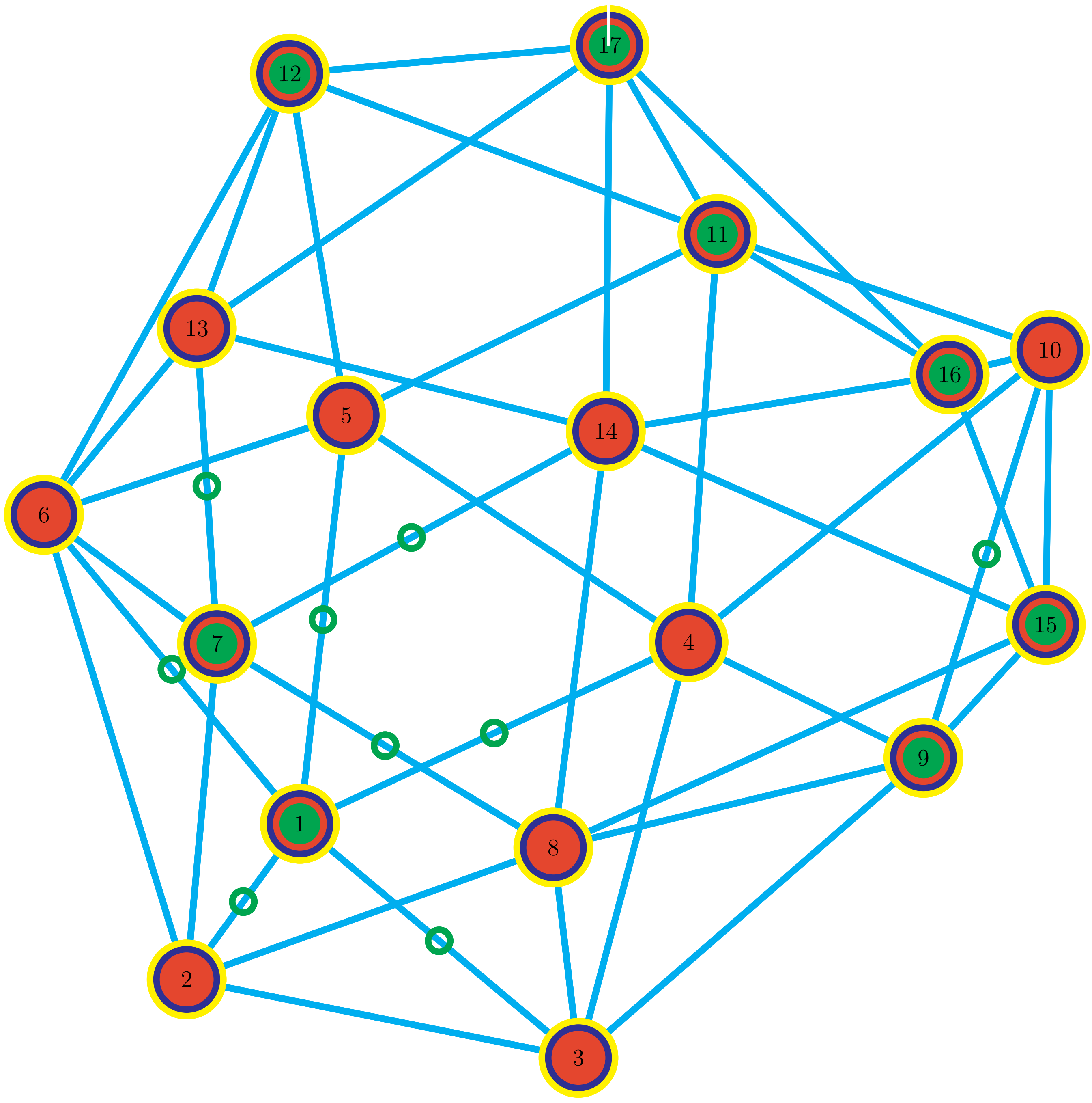

Figure 19. Instructions: 52: color edge 9–10 Green, 53: uncolor vertex 10 Green,

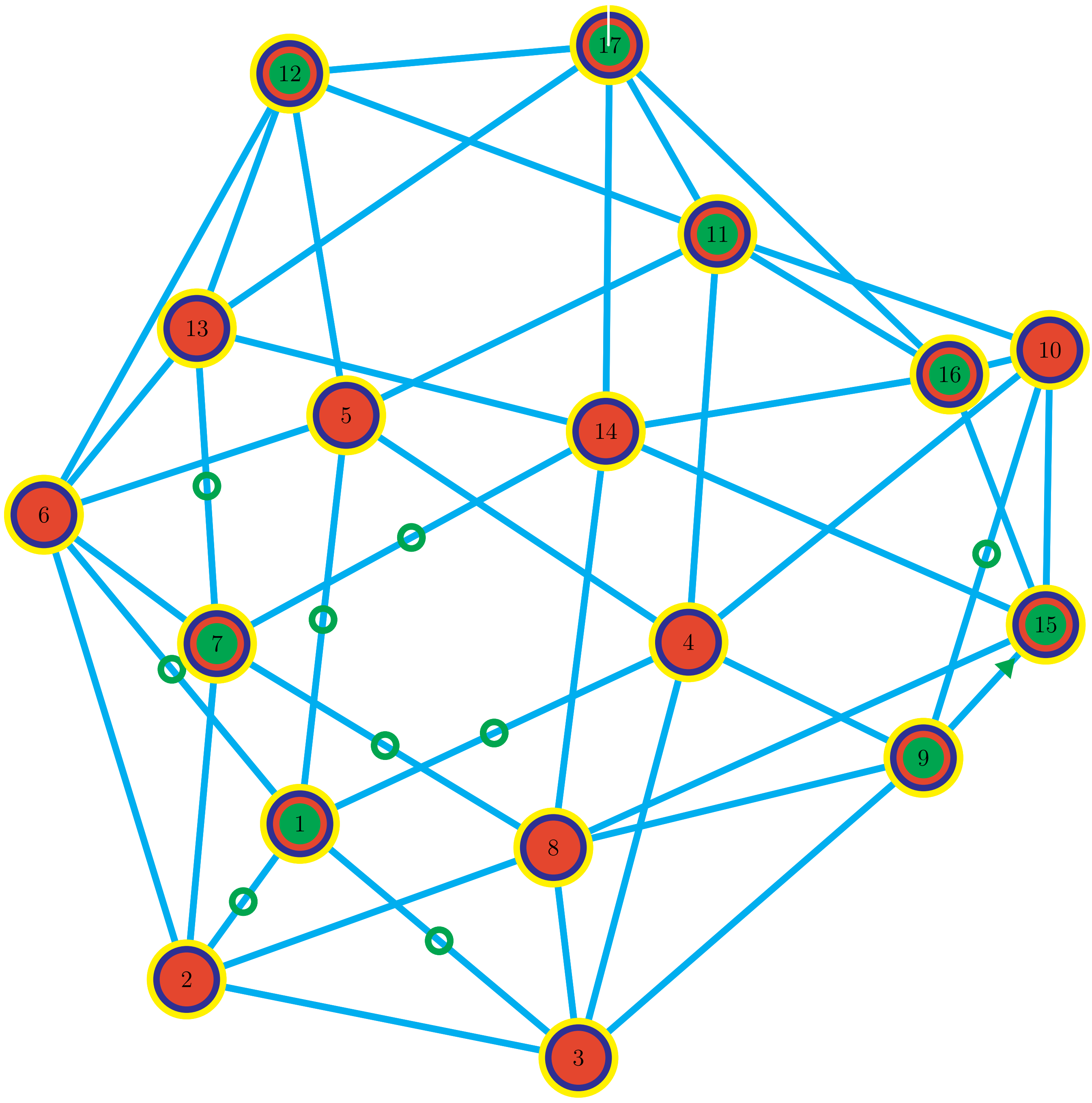

Figure 20. Instructions: 55: place edge 9→15 Green ArrowR,

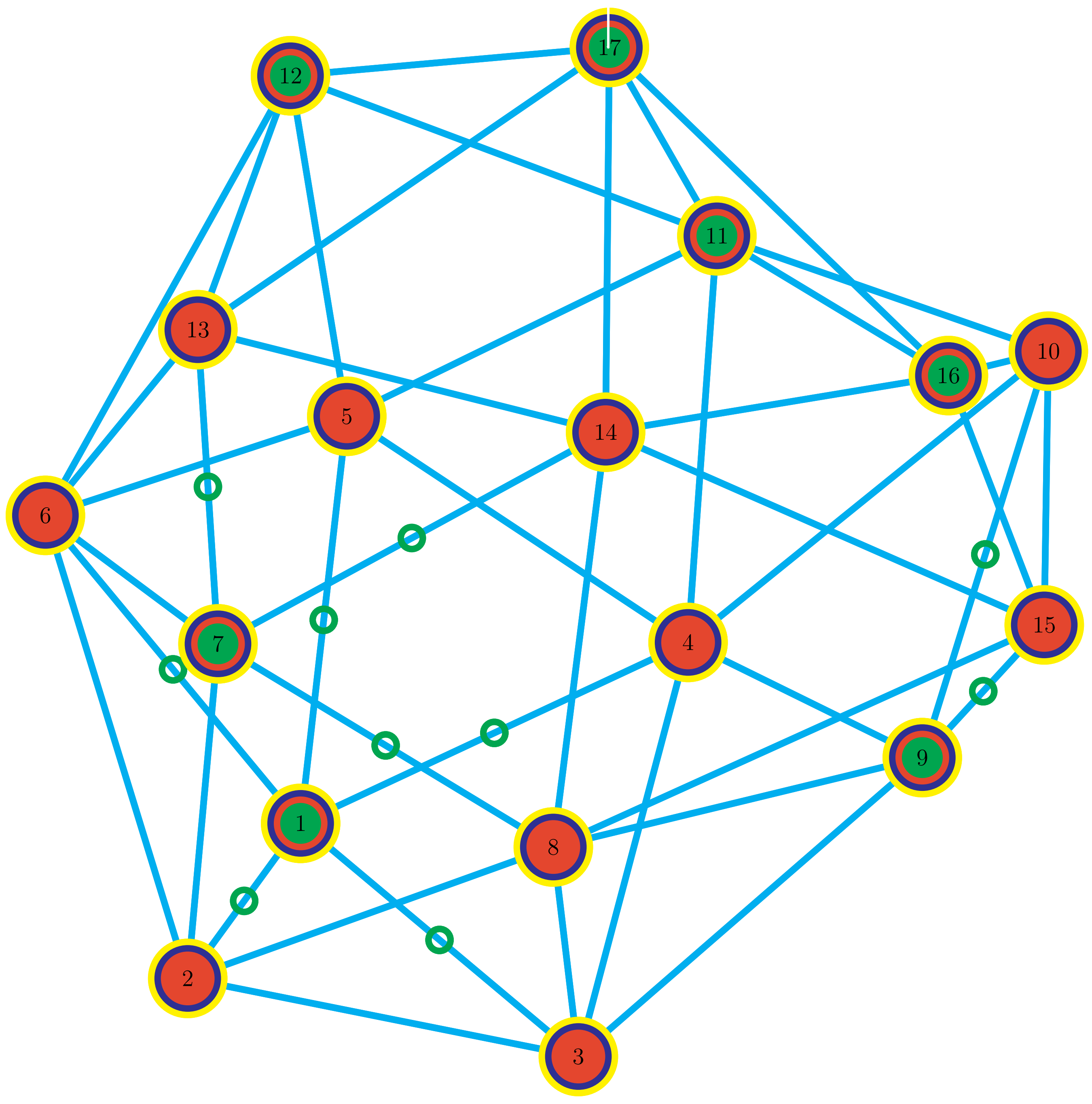

Figure 21. Instructions: 58: color edge 9–15 Green, 59: uncolor vertex 15 Green,

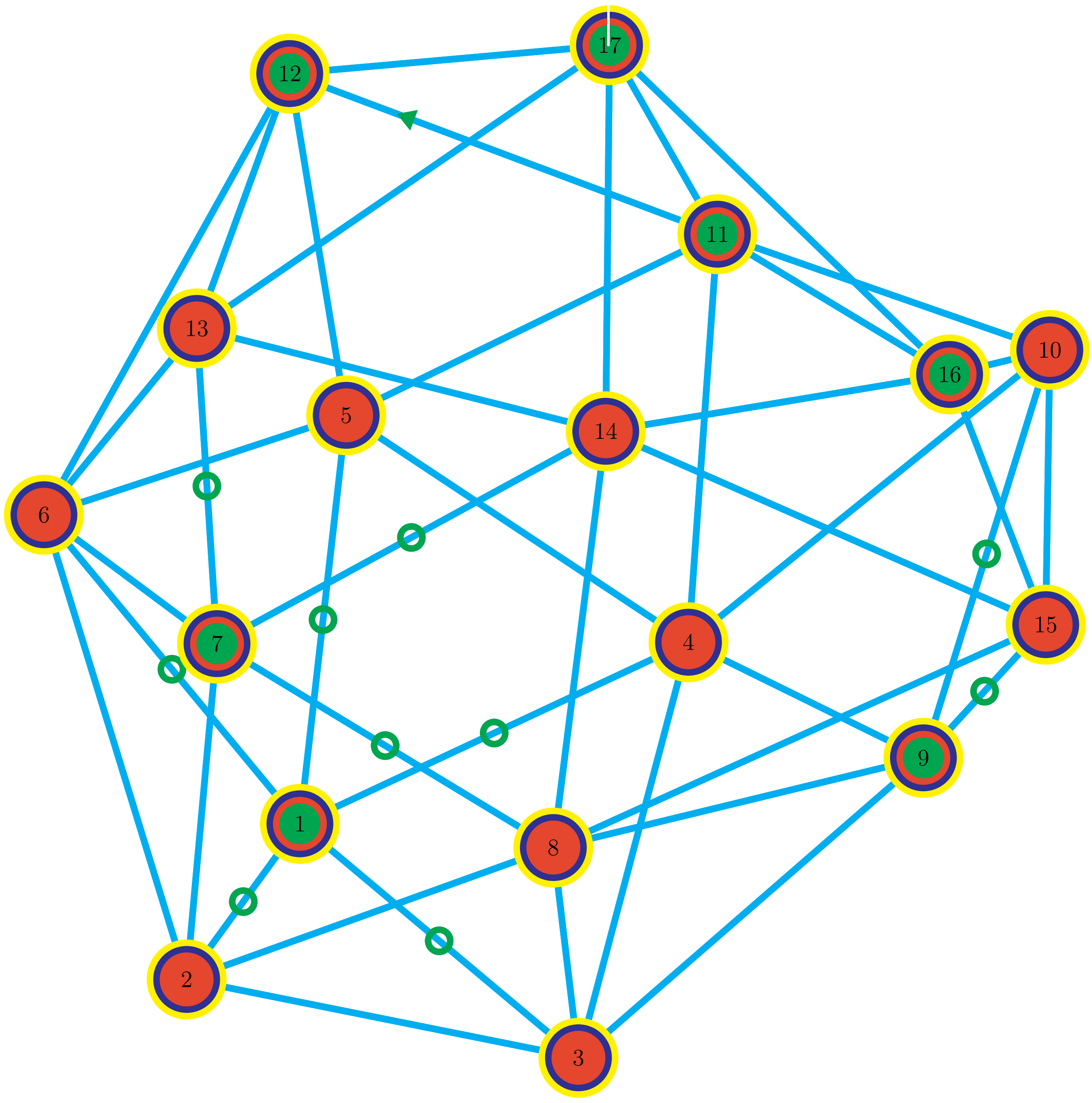

Figure 22. Instructions: 61: place edge 11→12 Green ArrowR,

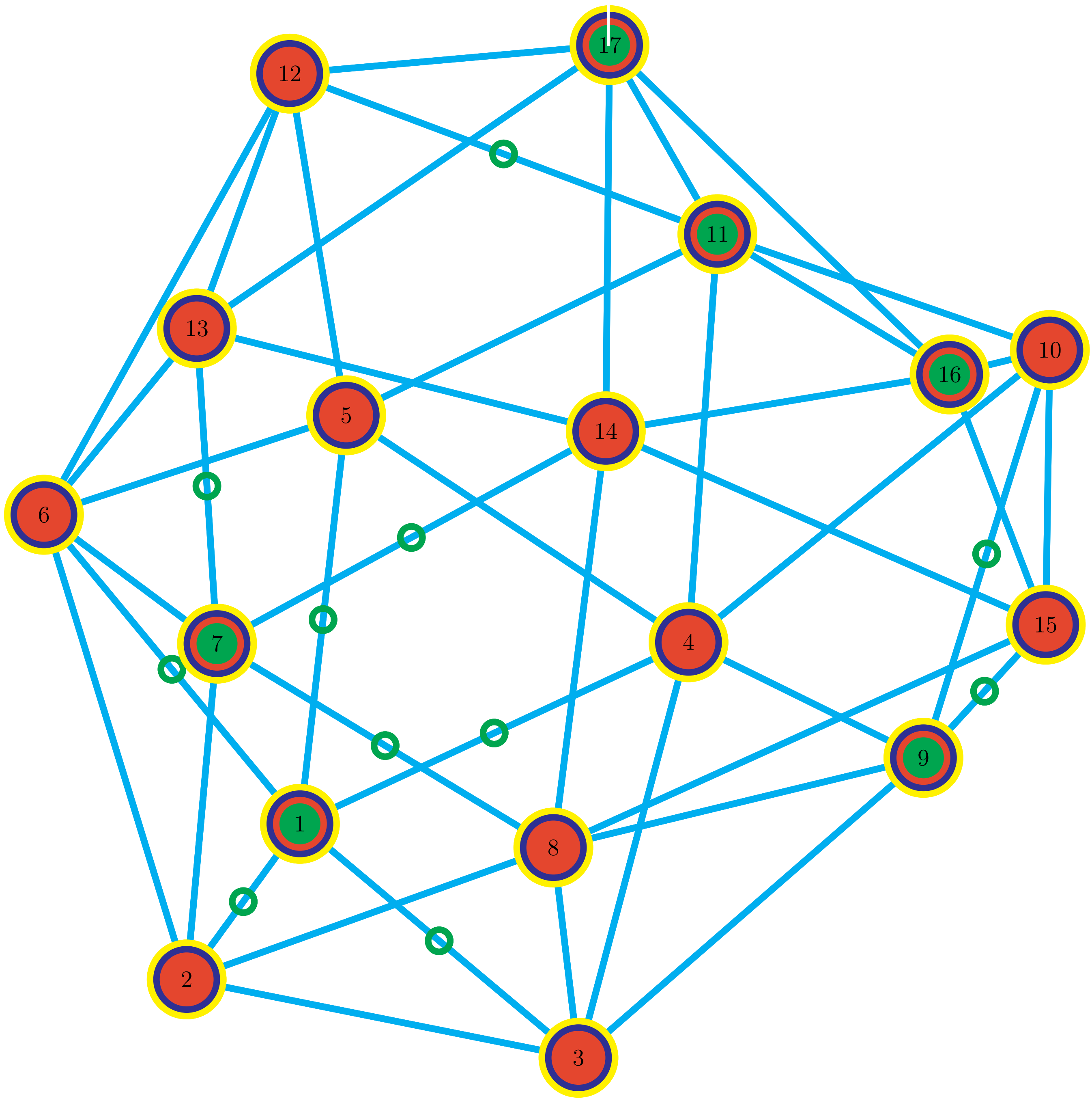

Figure 23. Instructions: 64: color edge 11–12 Green, 65: uncolor vertex 12 Green,

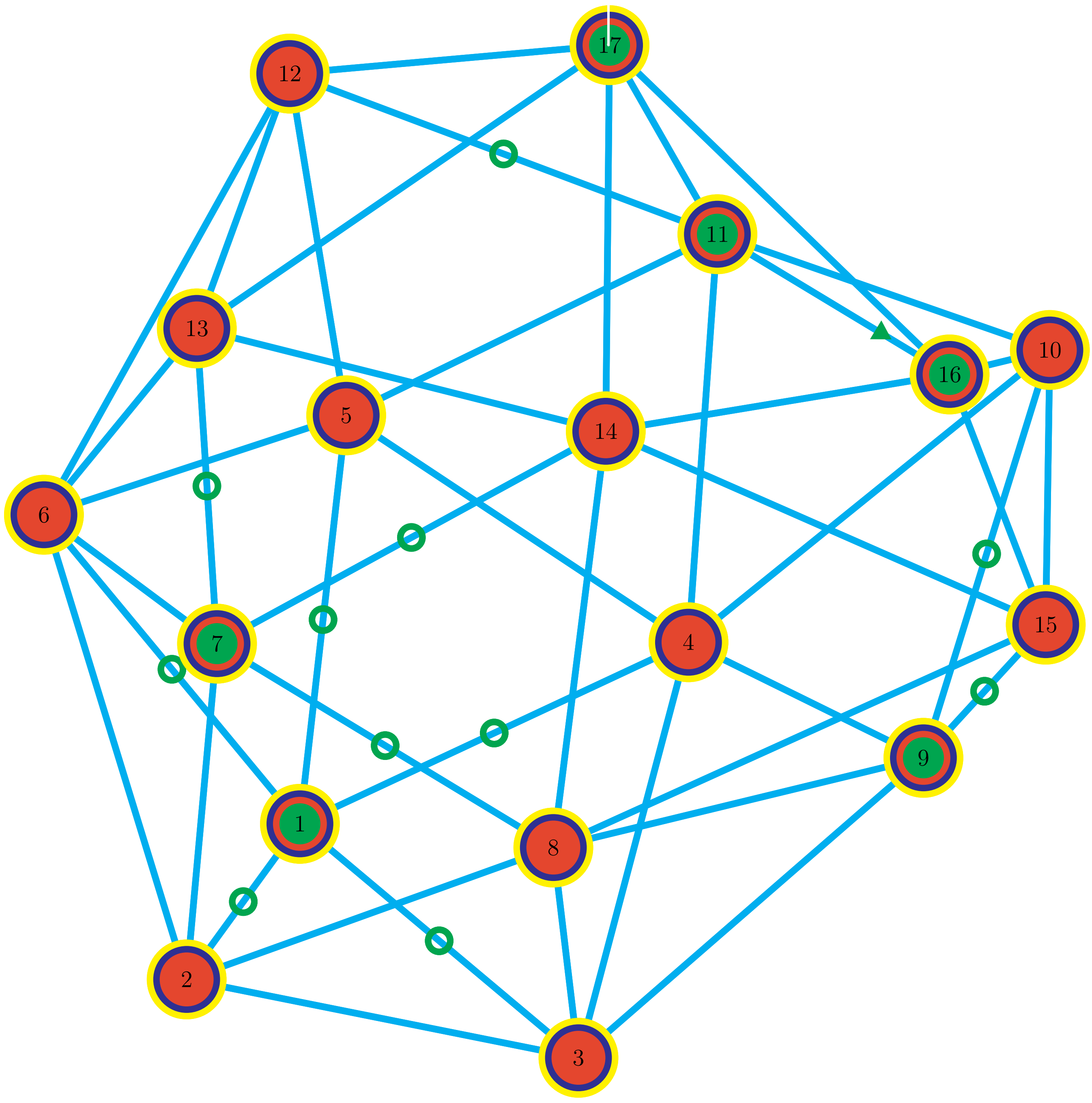

Figure 24. Instructions: 67: place edge 11→16 Green ArrowR,

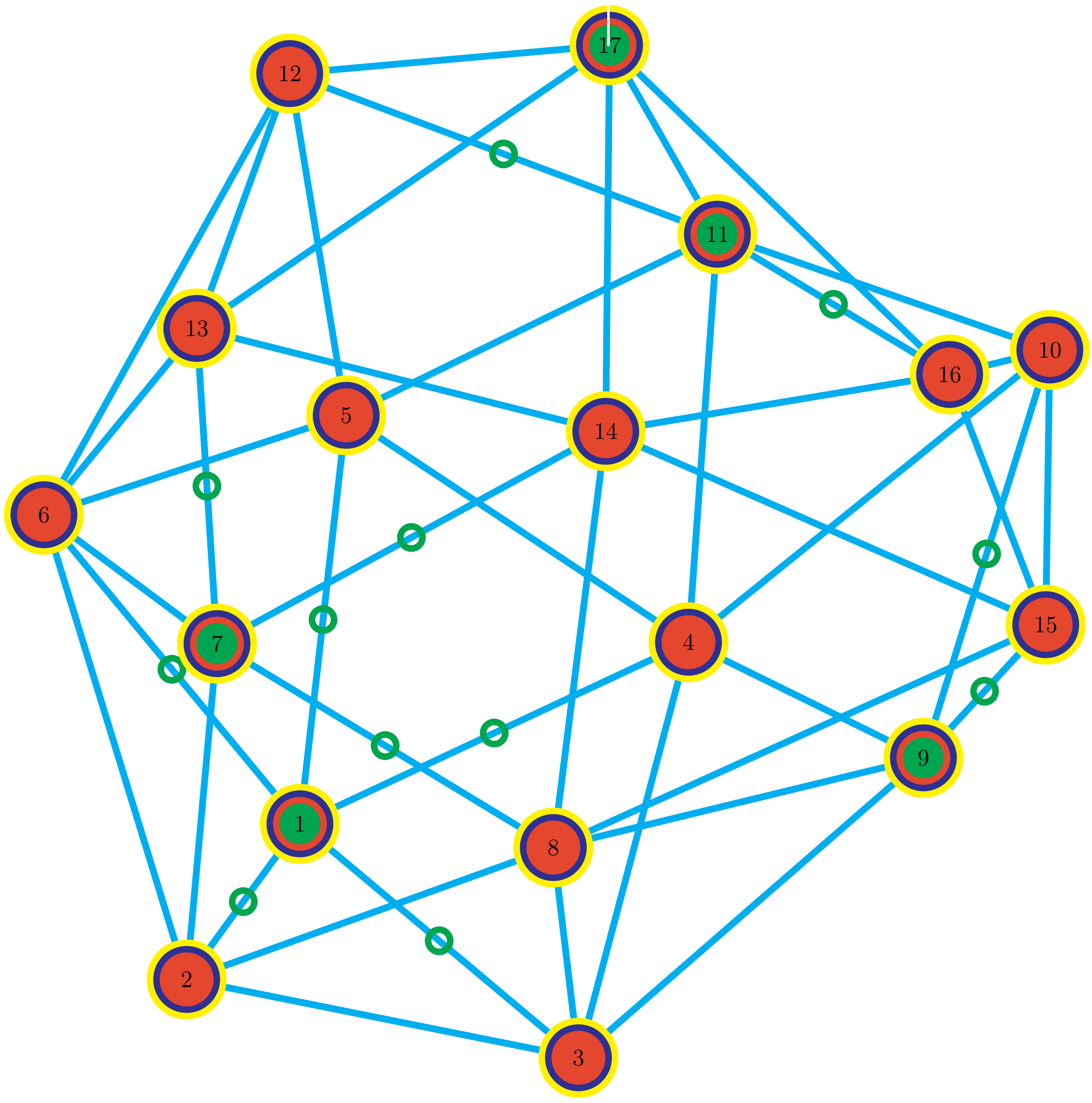

Figure 25. Instructions: 70: color edge 11–16 Green, 71: uncolor vertex 16 Green,

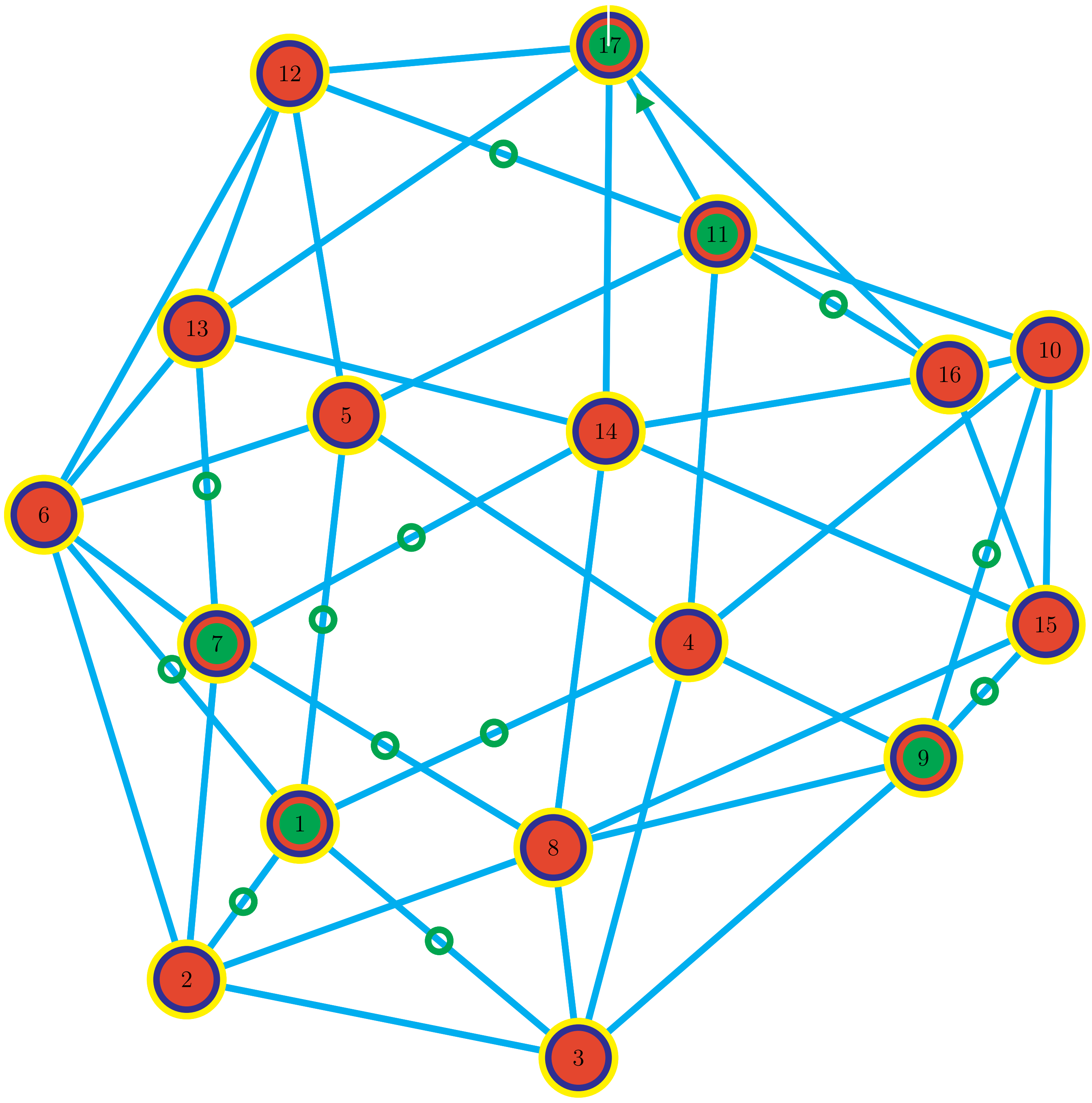

Figure 26. Instructions: 73: place edge 11→17 Green ArrowR,

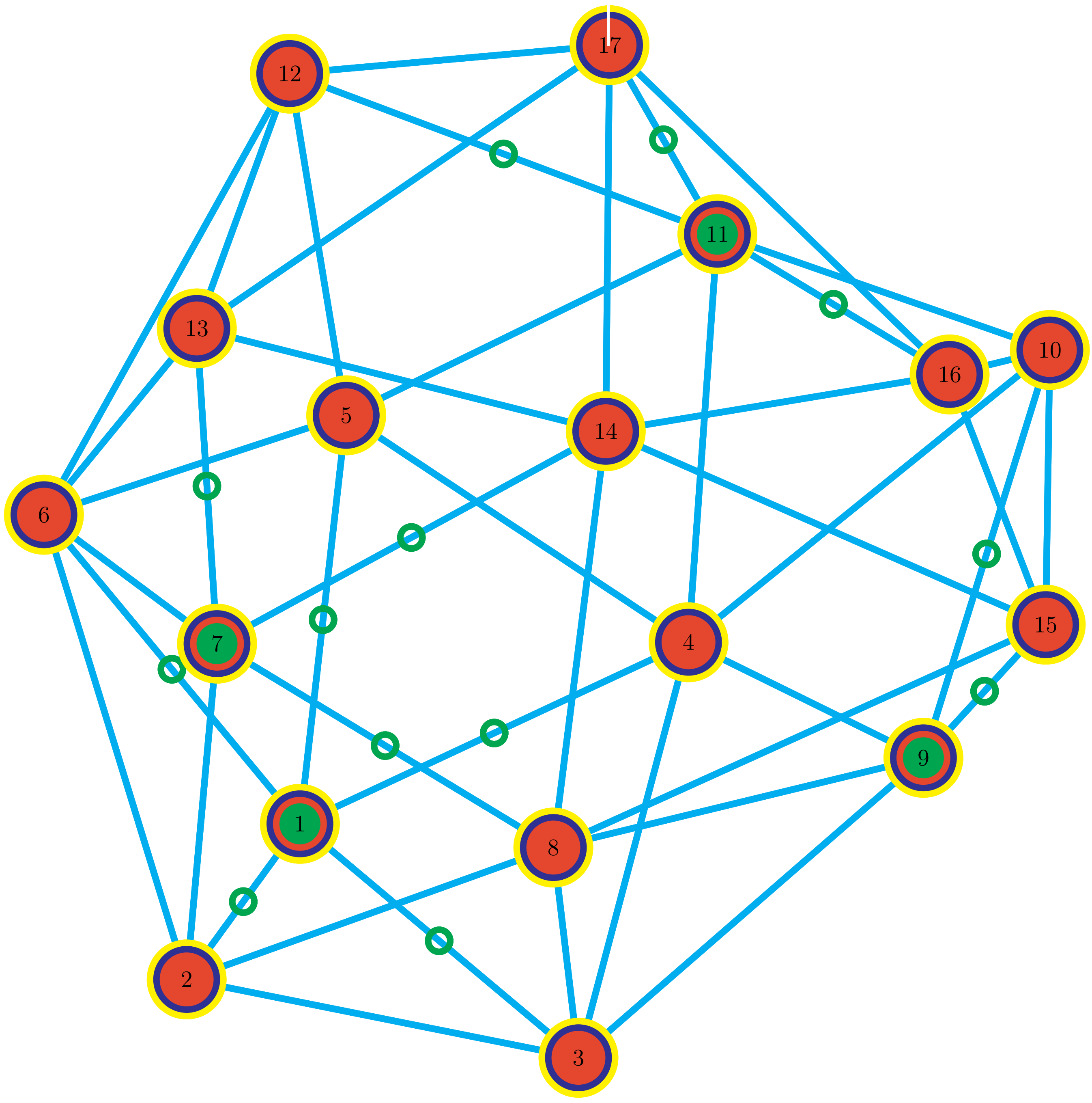

Figure 27. Instructions: 76: color edge 11–17 Green, 77: uncolor vertex 17 Green,

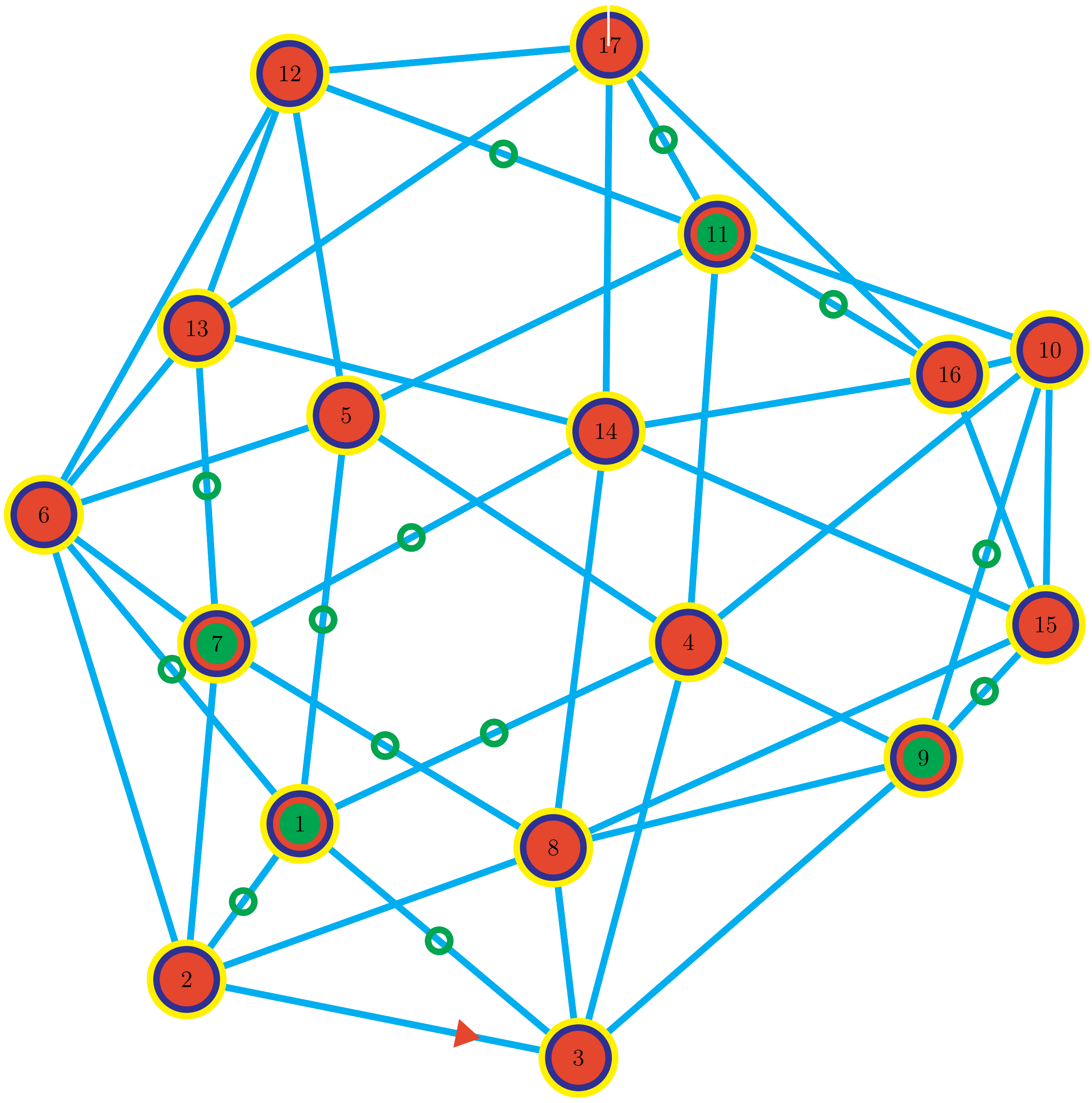

Figure 28. Instructions: 79: place edge 2→3 Red ArrowR,

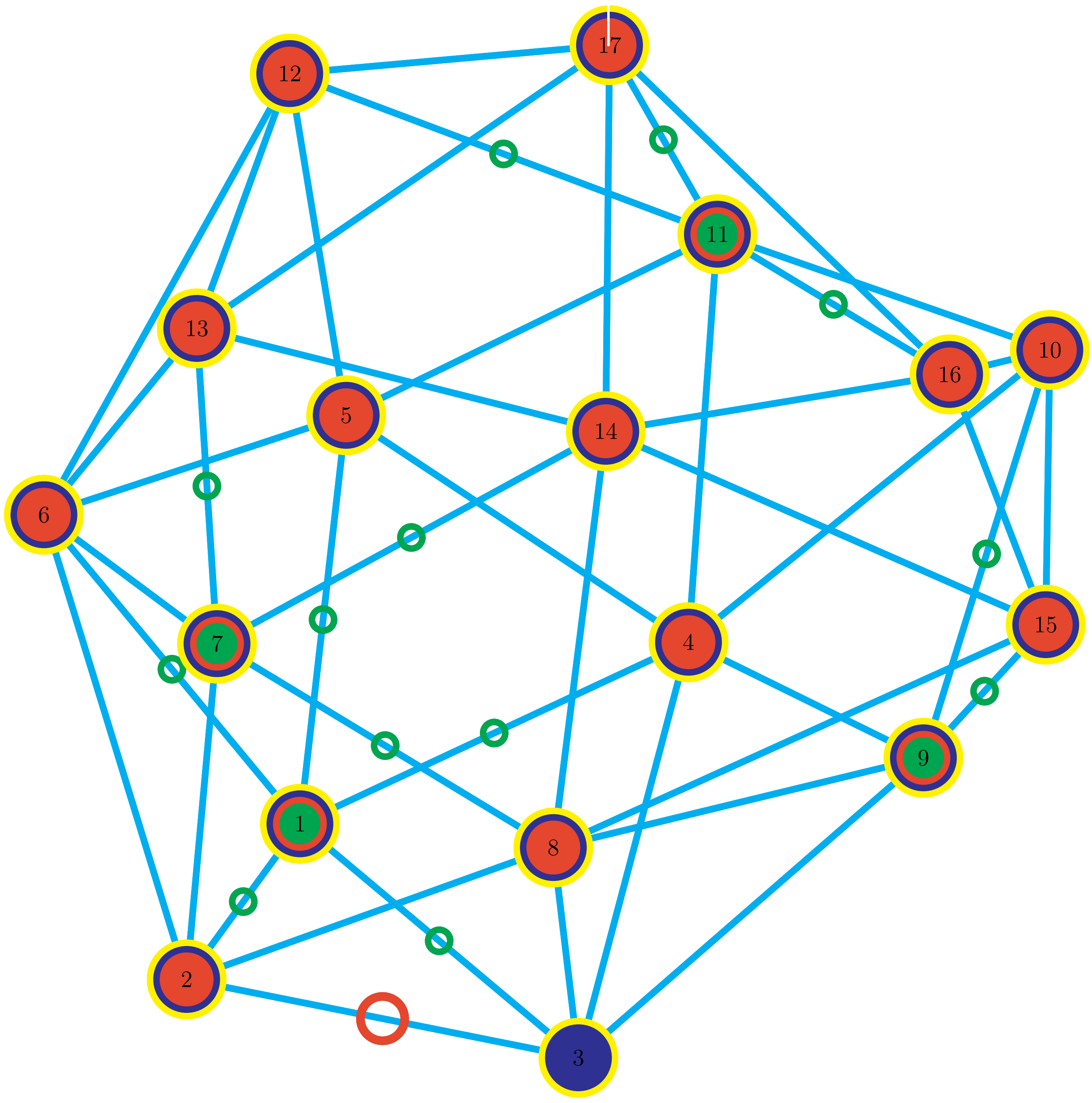

Figure 29. Instructions: 82: color edge 2–3 Red, 83: uncolor vertex 3 Red,

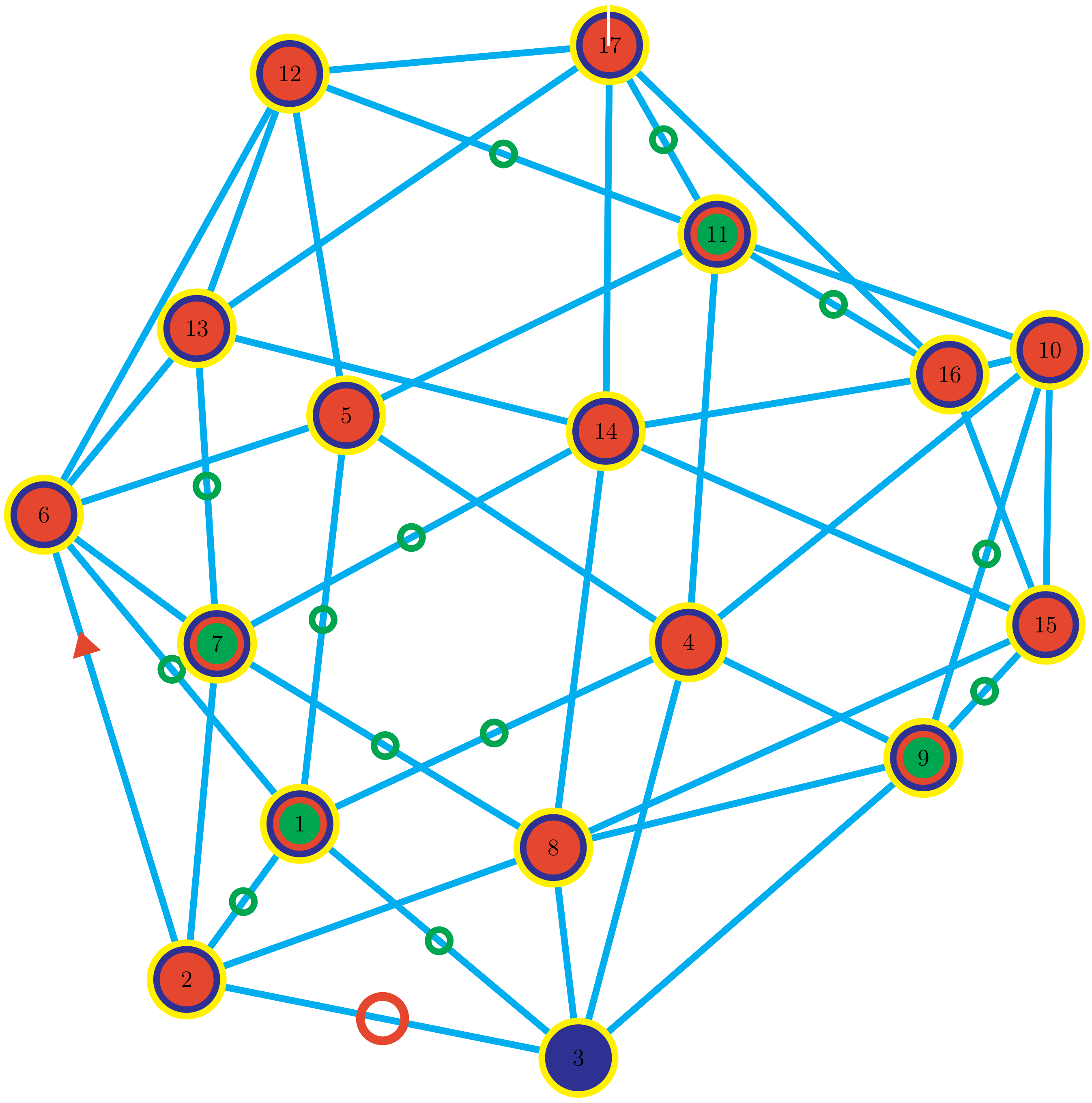

Figure 30. Instructions: 85: place edge 2→6 Red ArrowR,

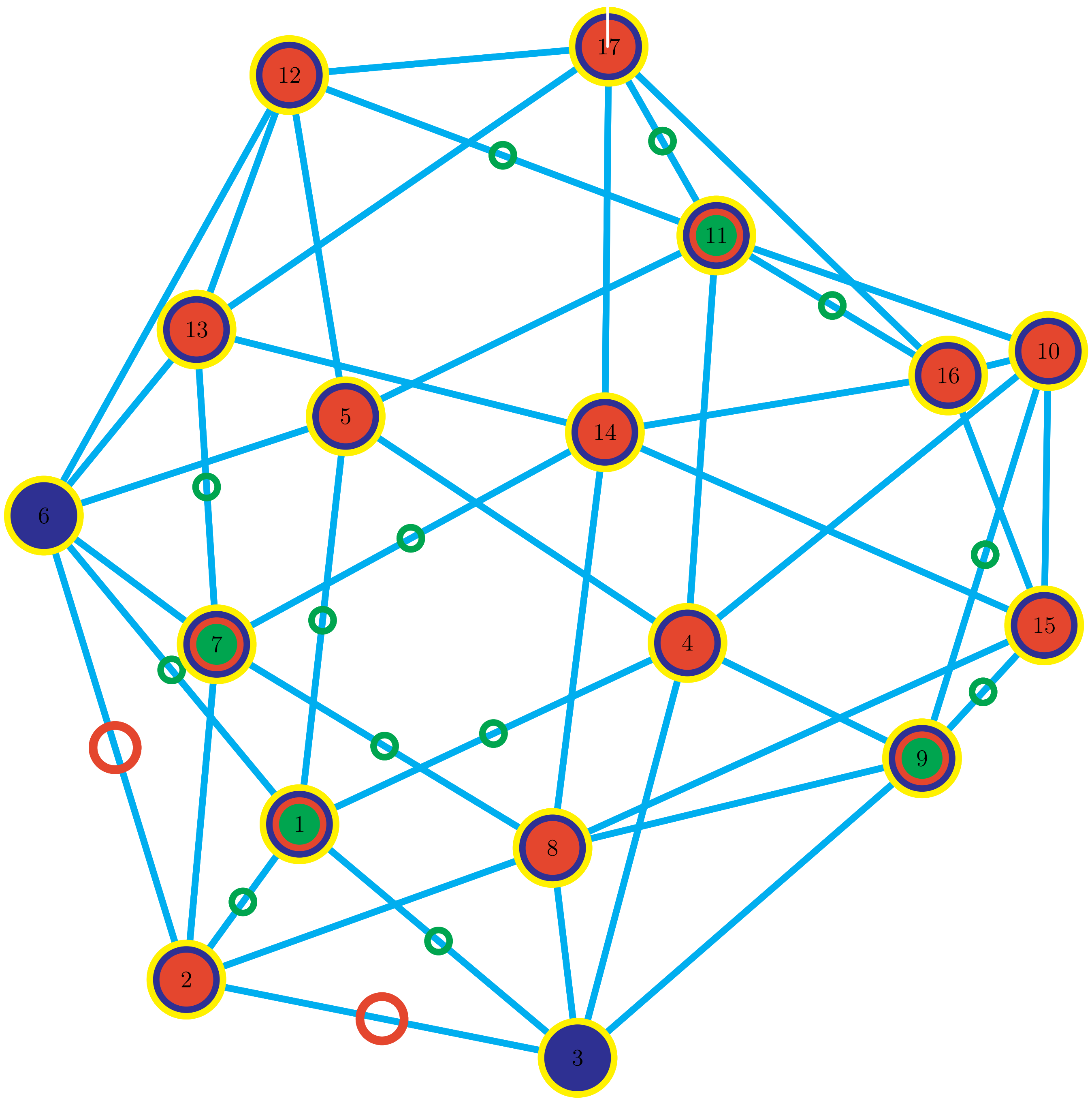

Figure 31. Instructions: 88: color edge 2–6 Red, 89: uncolor vertex 6 Red,

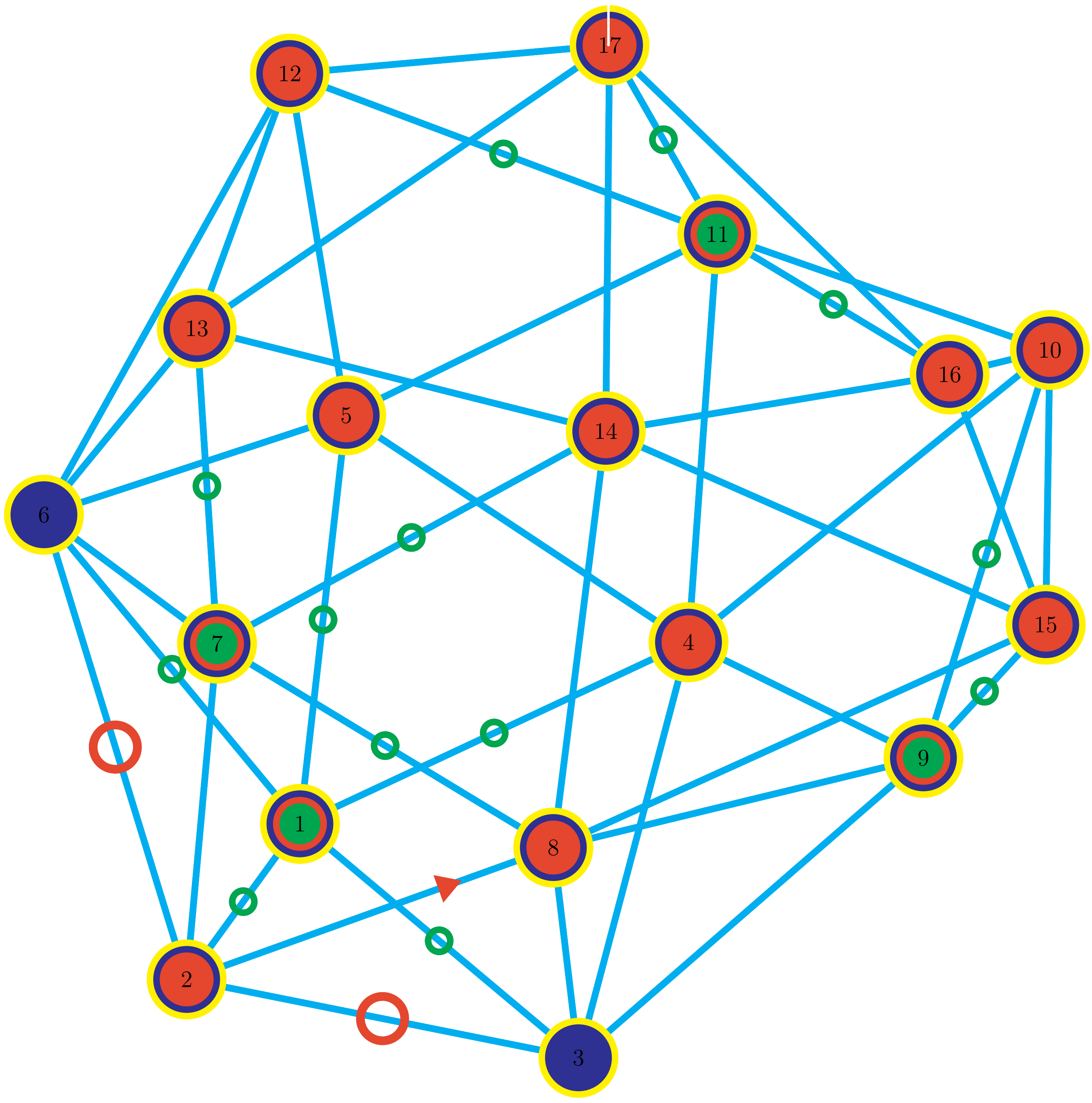

Figure 32. Instructions: 91: place edge 2→8 Red ArrowR,

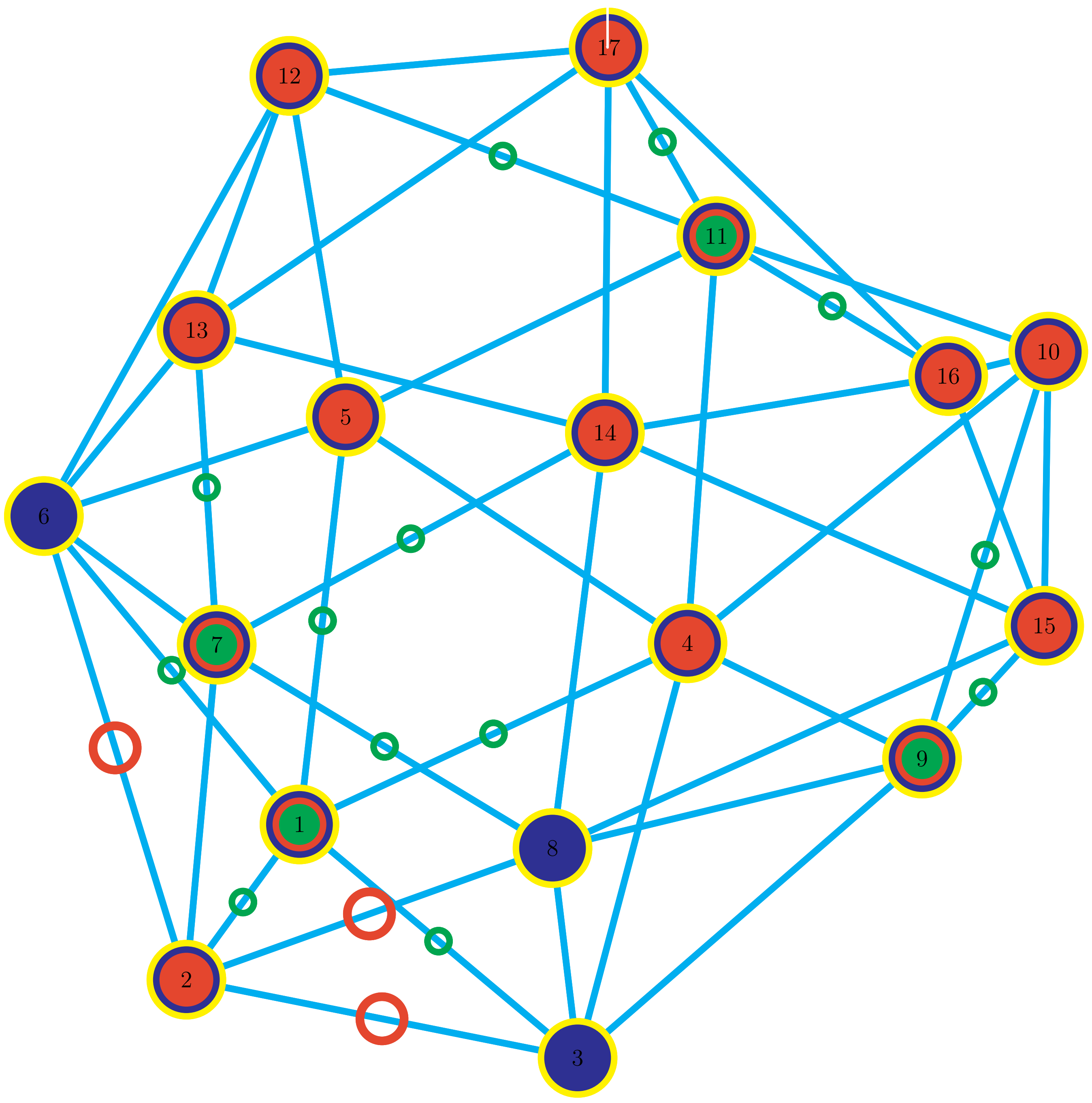

Figure 33. Instructions: 94: color edge 2–8 Red, 95: uncolor vertex 8 Red,

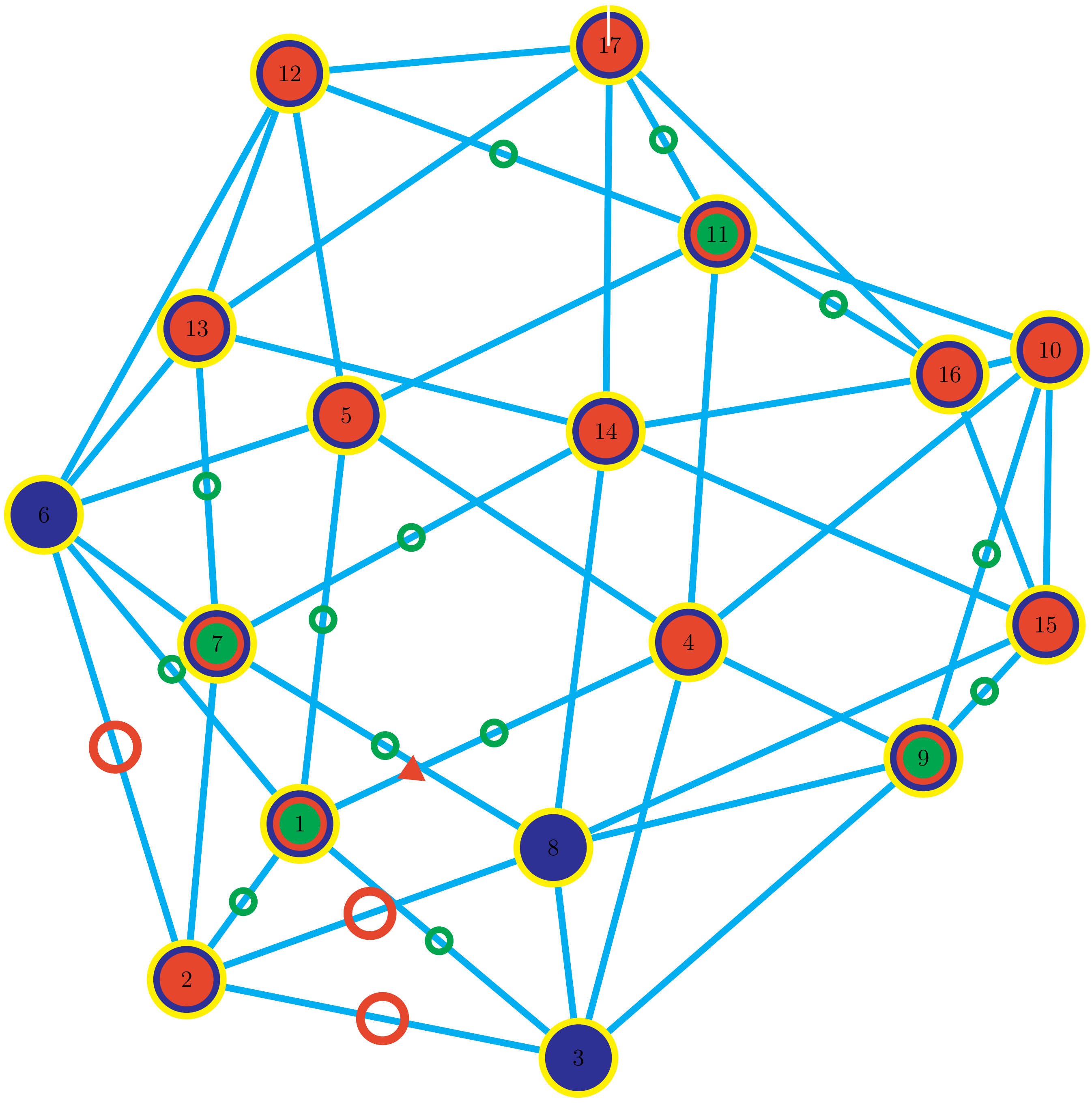

Figure 34. Instructions: 97: place edge 4→1 Red ArrowR,

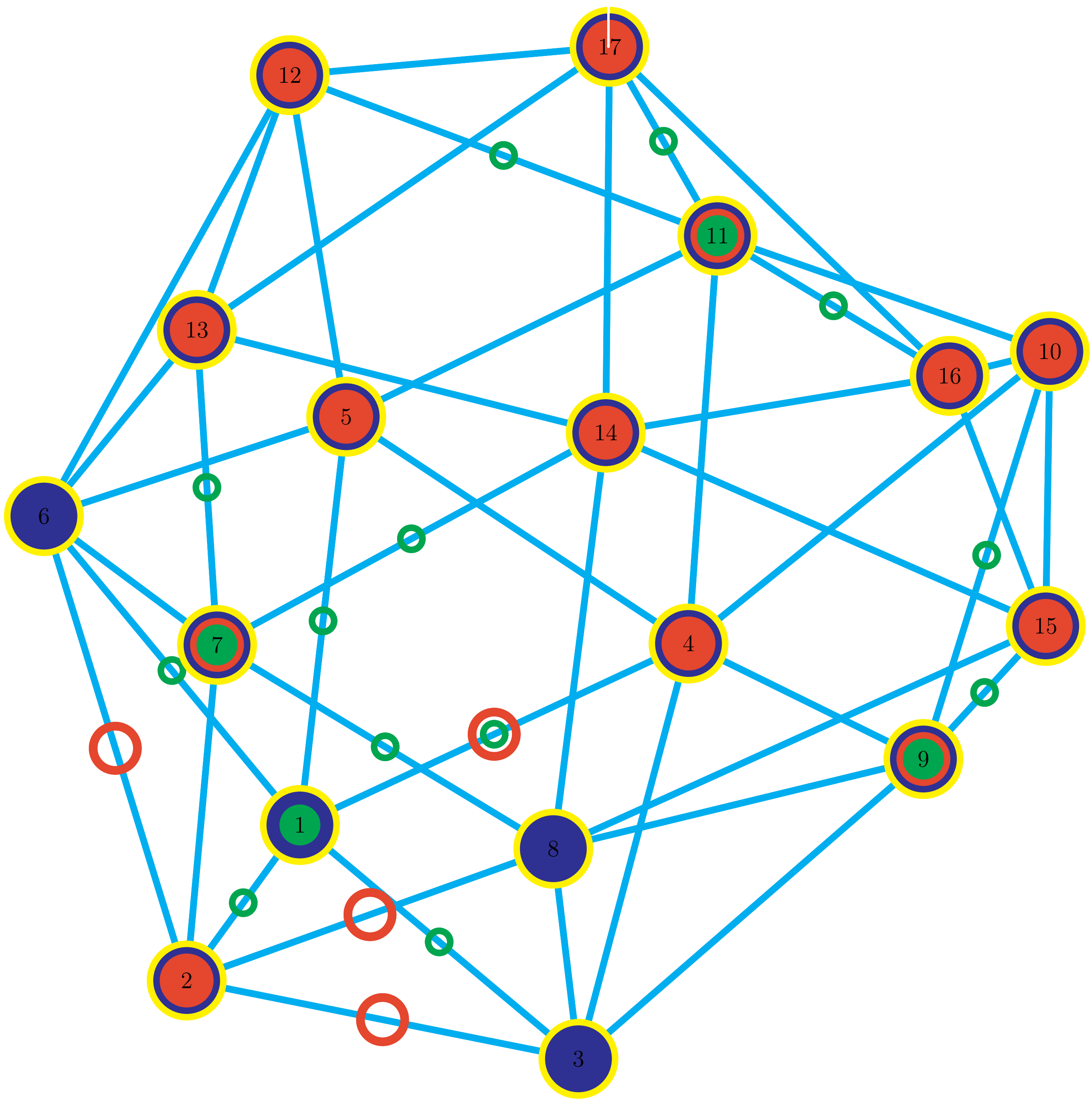

Figure 35. Instructions: 100: color edge 4–1 Red, 101: uncolor vertex 1 Red,

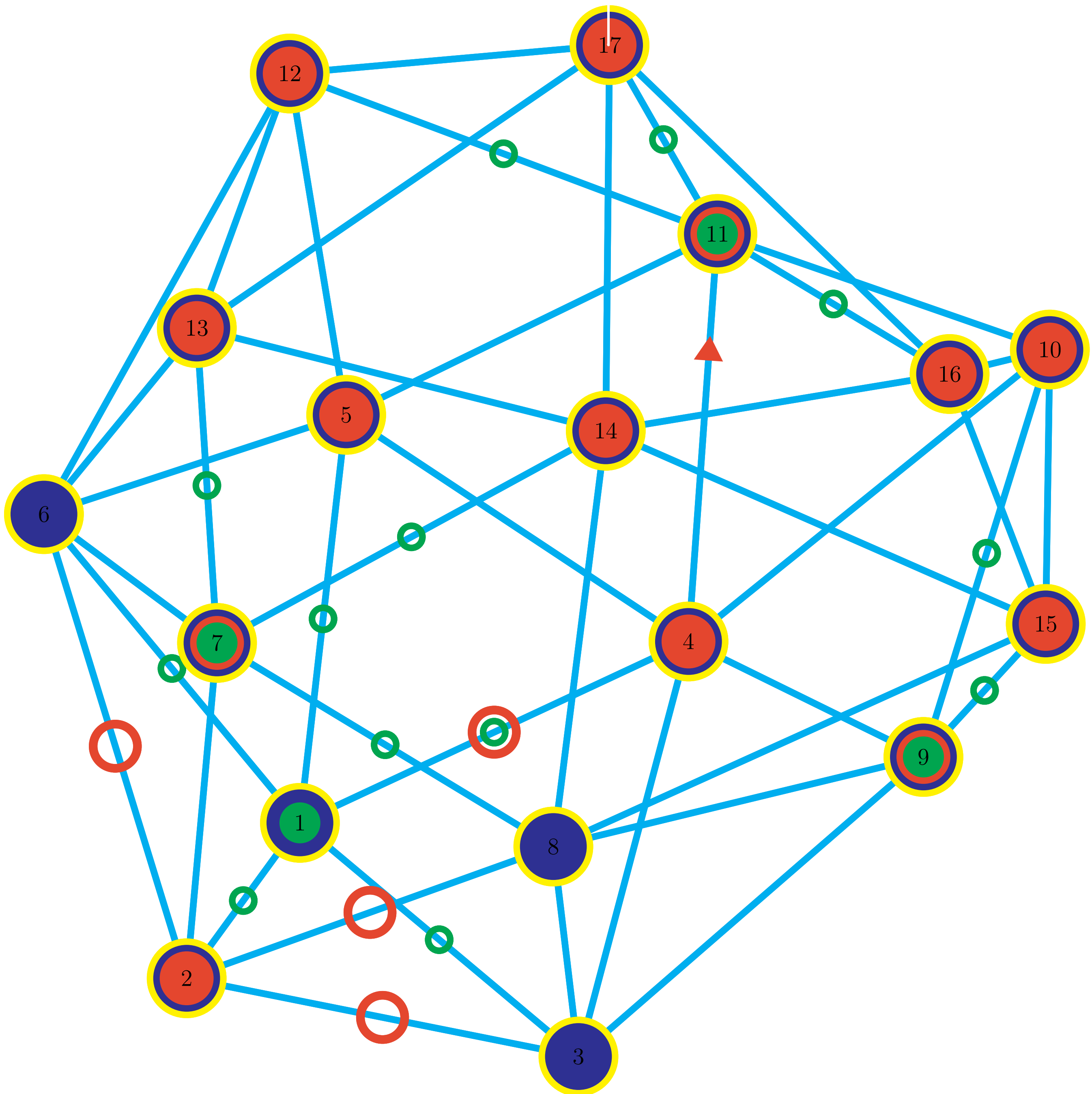

Figure 36. Instructions: 103: place edge 4→11 Red ArrowR,

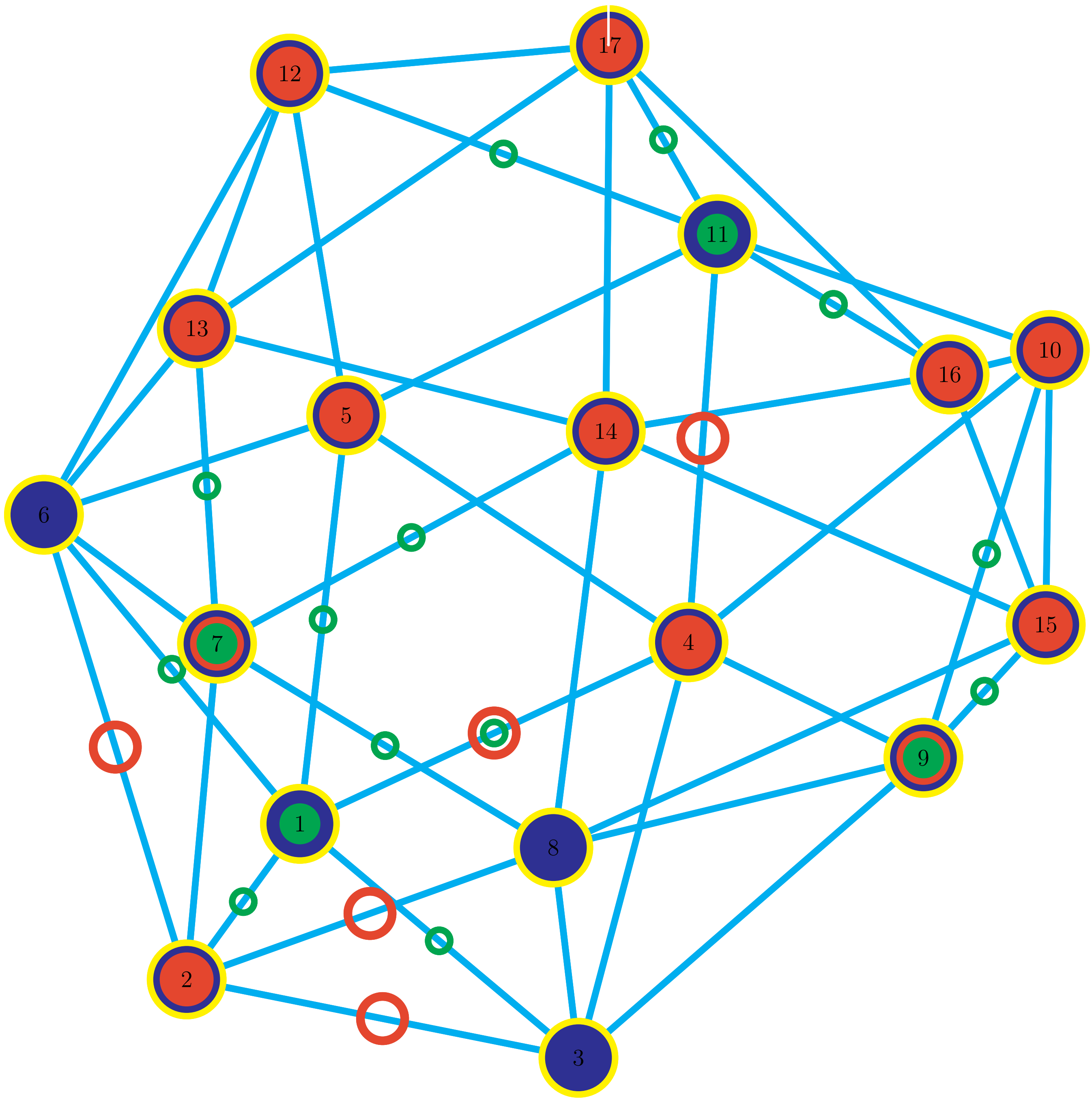

Figure 37. Instructions: 106: color edge 4–11 Red, 107: uncolor vertex 11 Red,

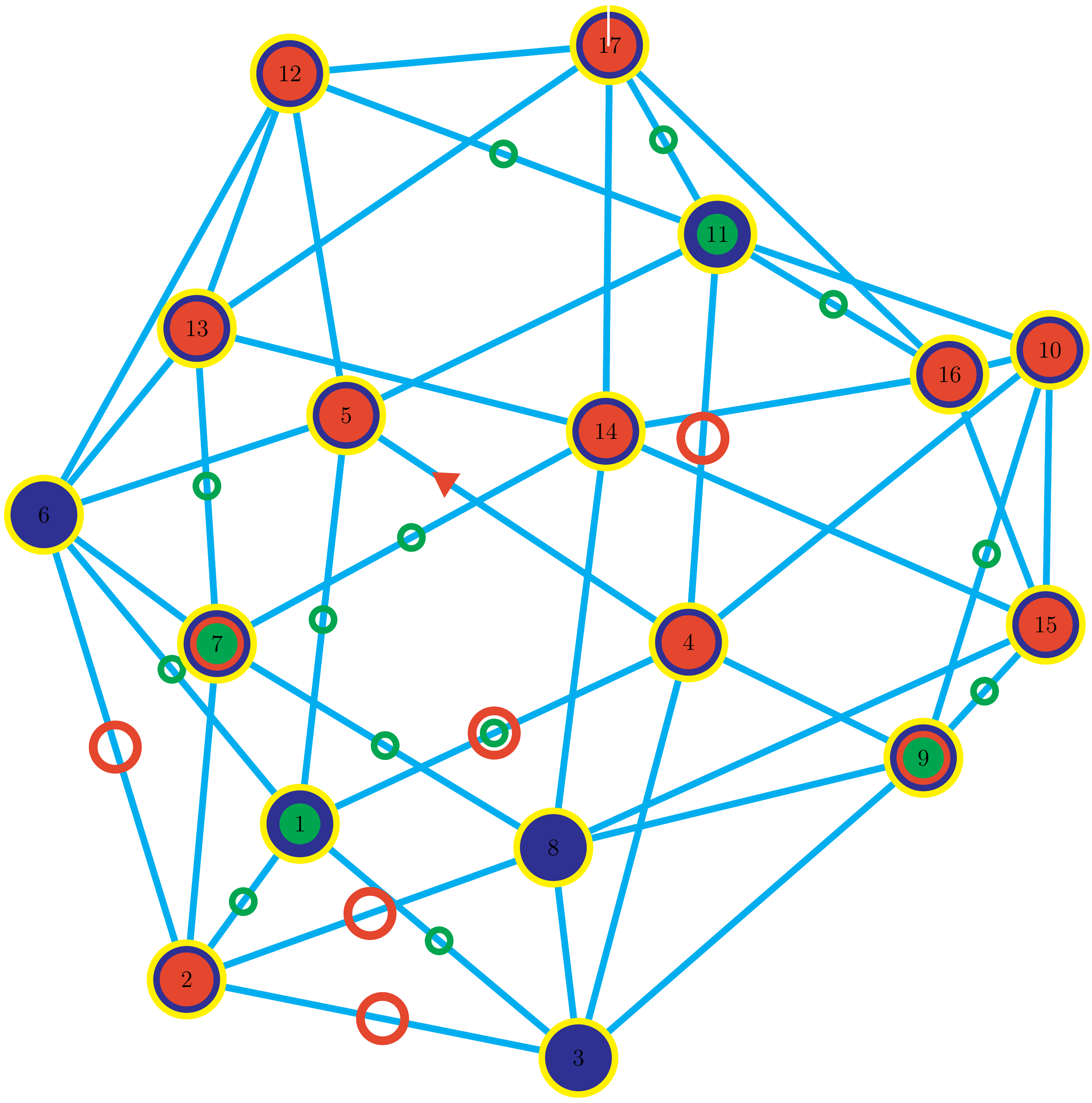

Figure 38. Instructions: 109: place edge 4→5 Red ArrowR,

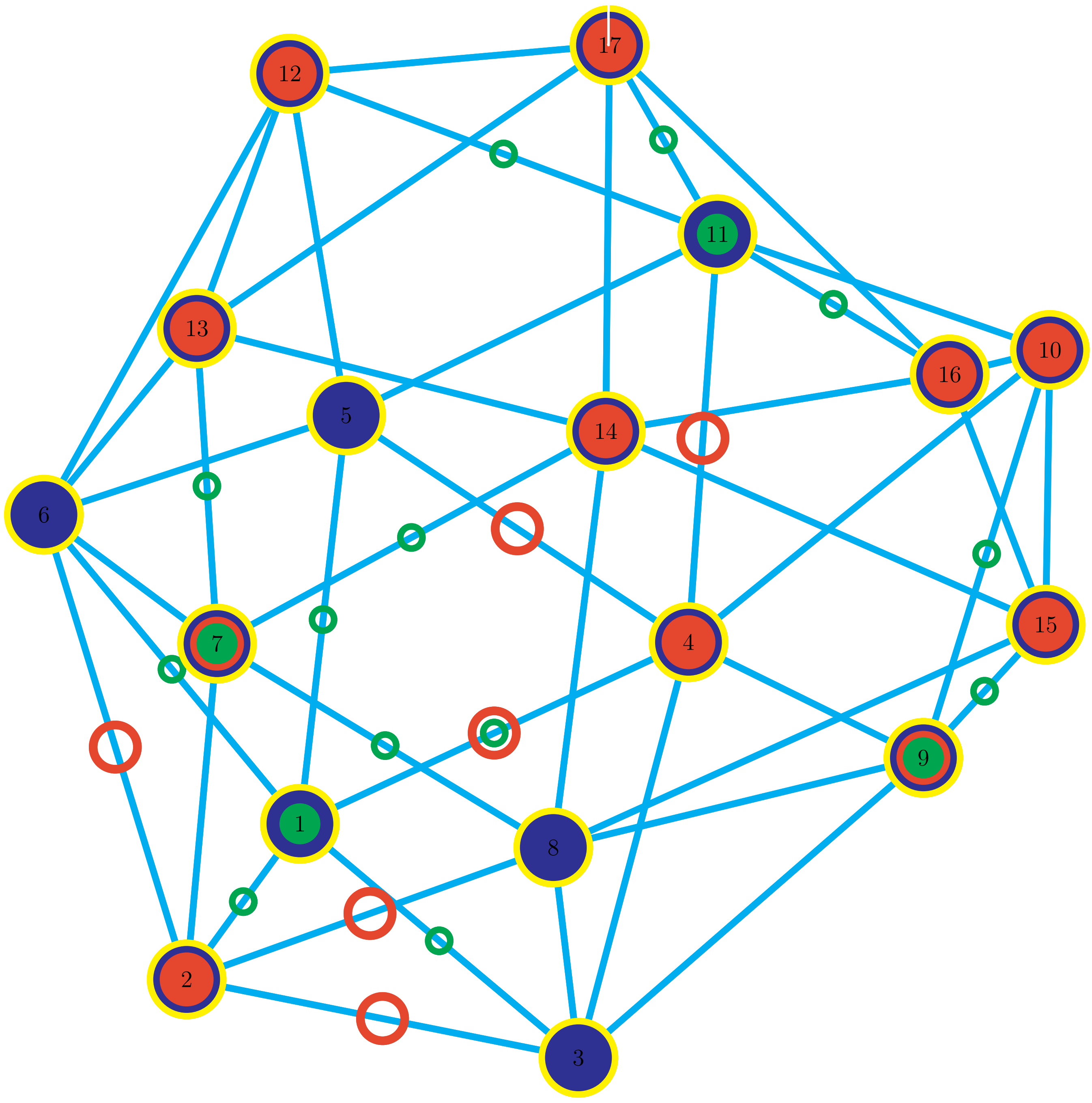

Figure 39. Instructions: 112: color edge 4–5 Red, 113: uncolor vertex 5 Red,

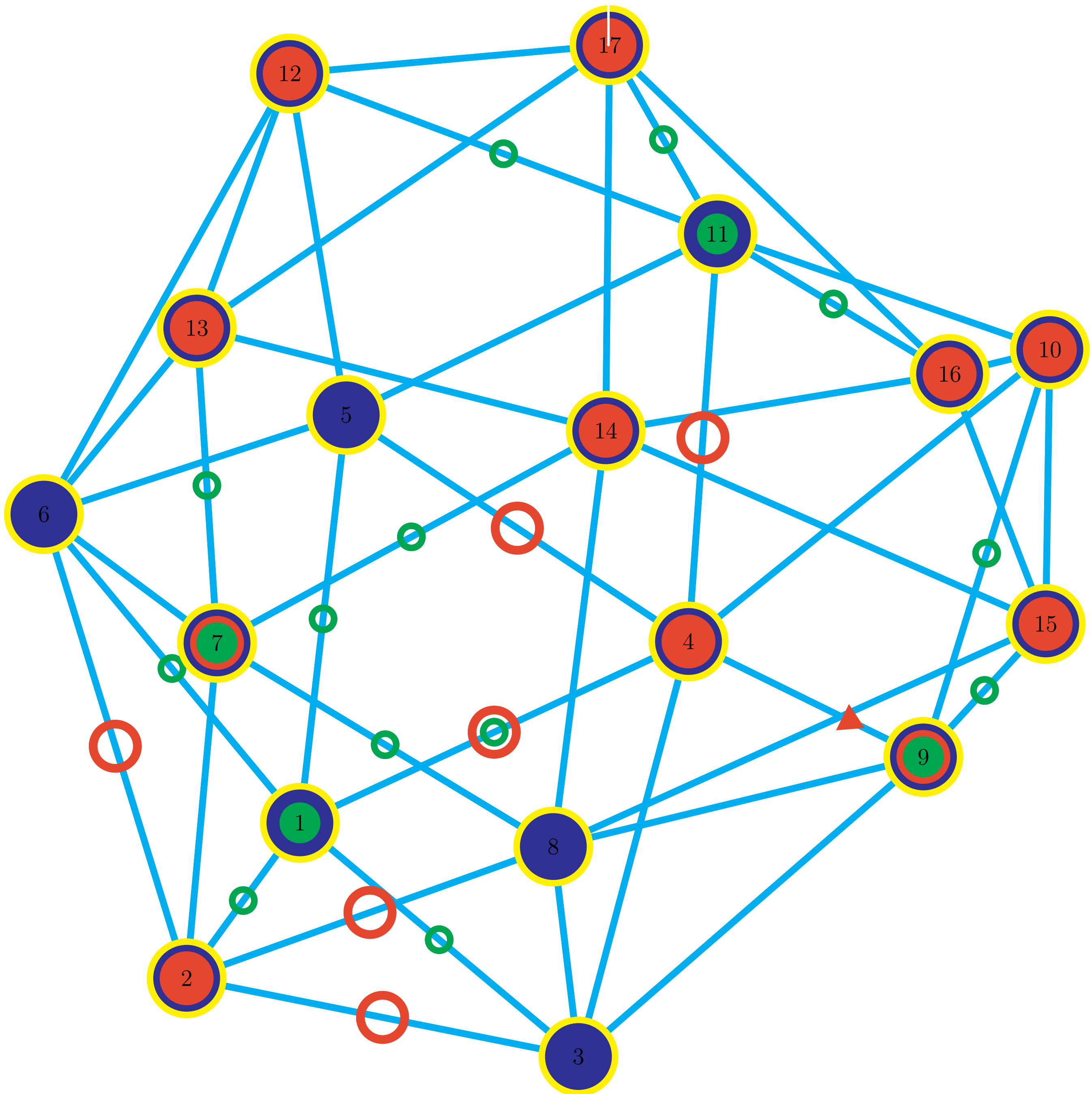

Figure 40. Instructions: 115: place edge 4→9 Red ArrowR,

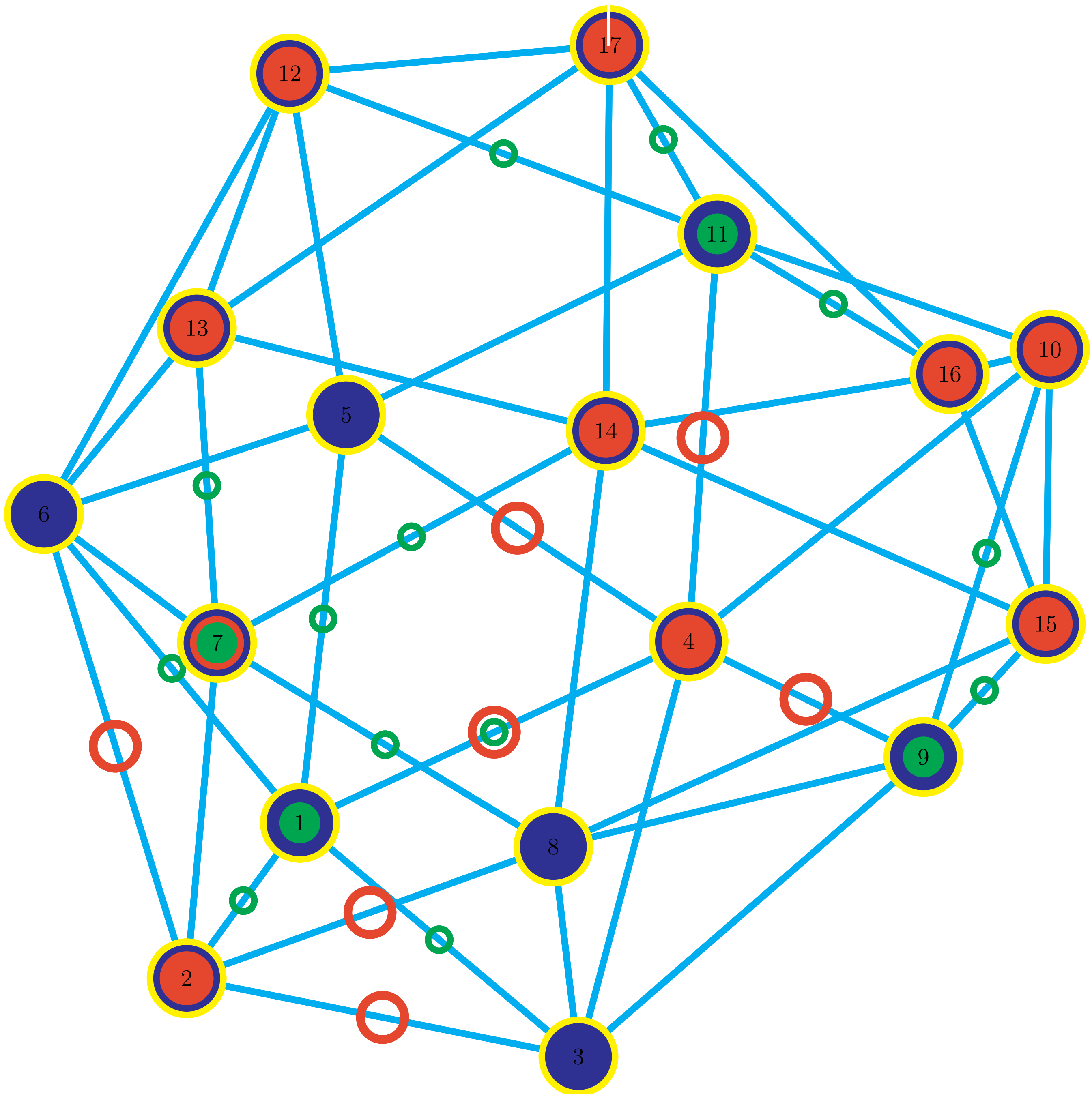

Figure 41. Instructions: 118: color edge 4–9 Red, 119: uncolor vertex 9 Red,

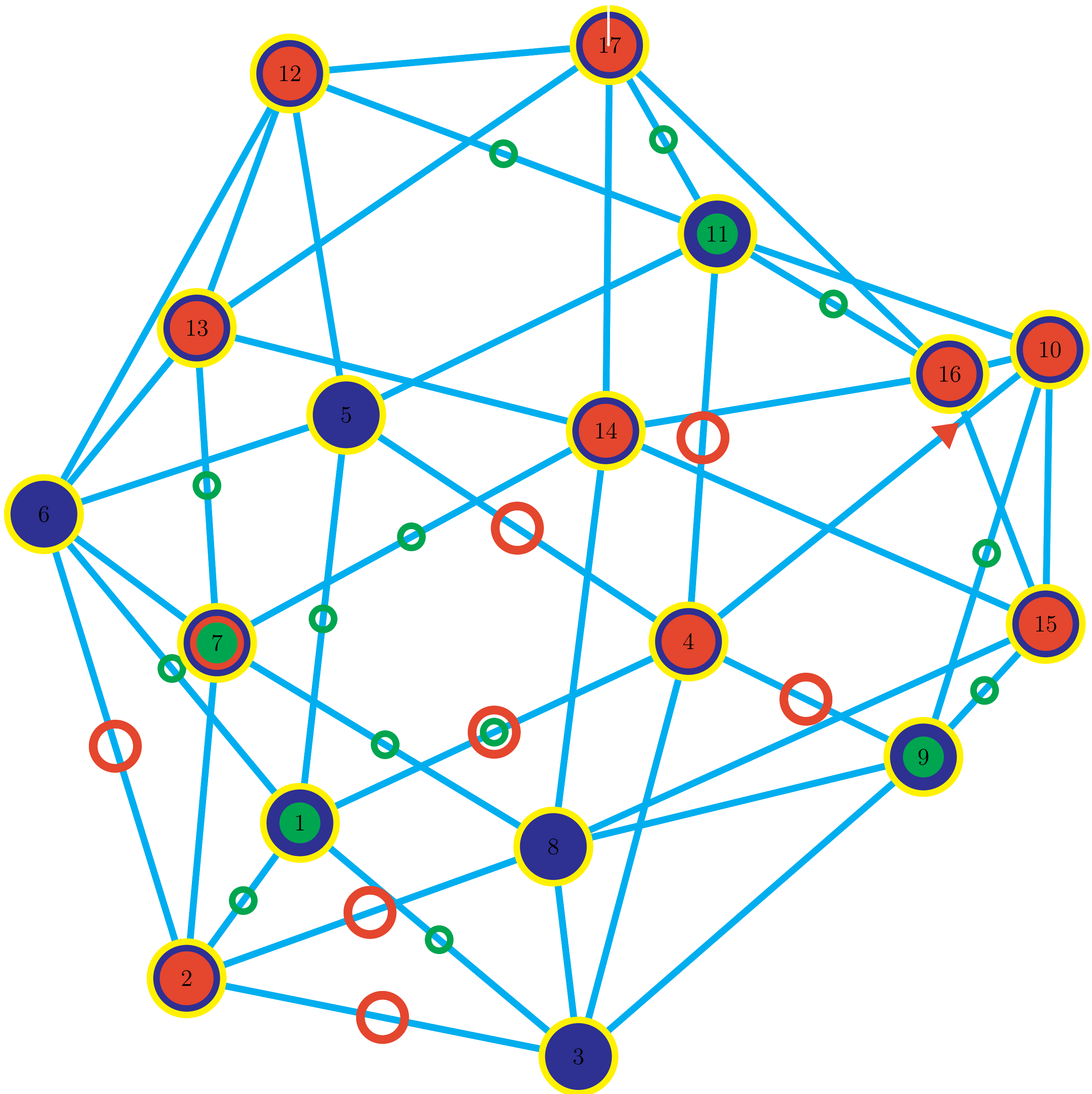

Figure 42. Instructions: 121: place edge 4→10 Red ArrowR,

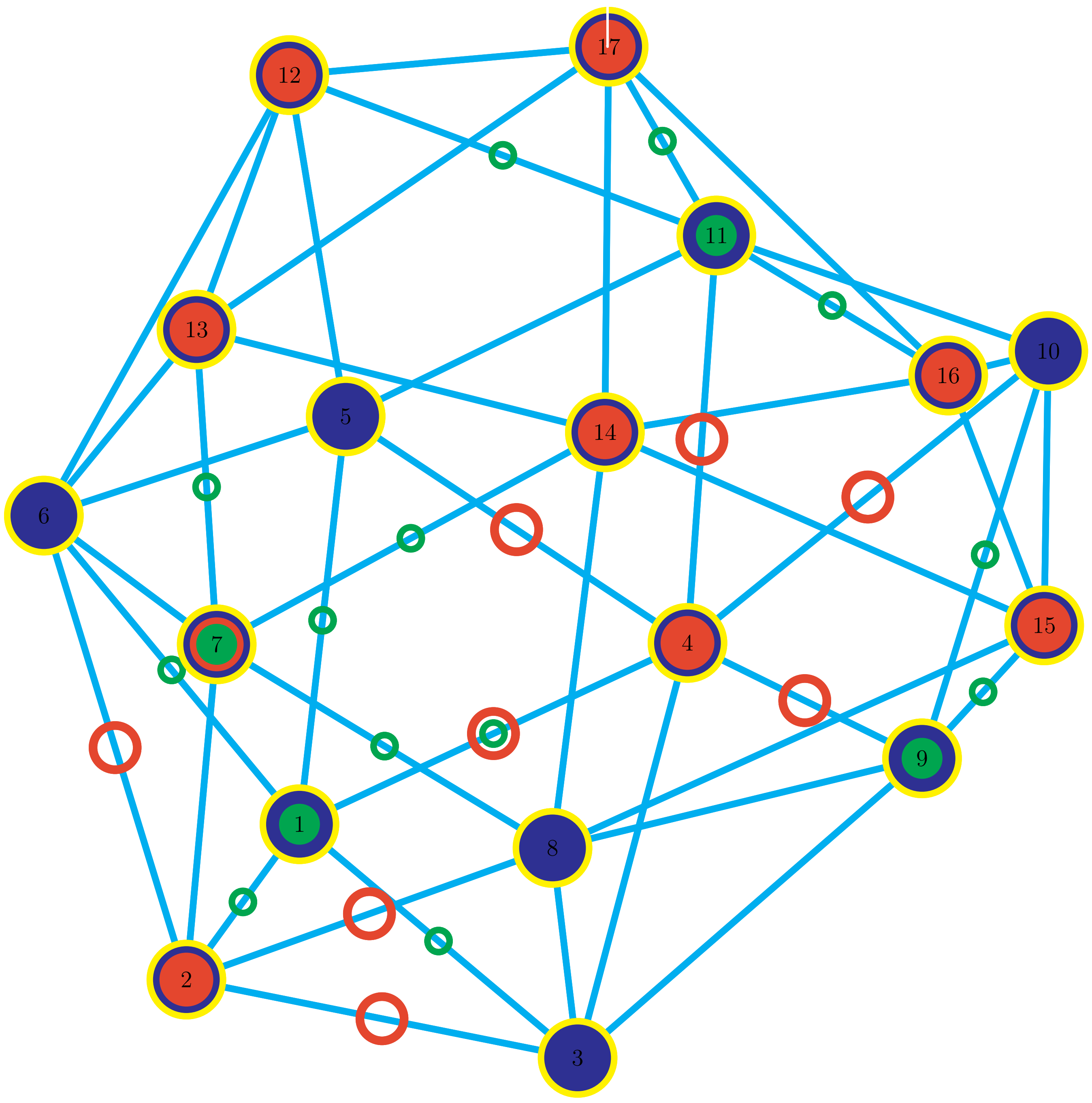

Figure 43. Instructions: 124: color edge 4–10 Red, 125: uncolor vertex 10 Red,

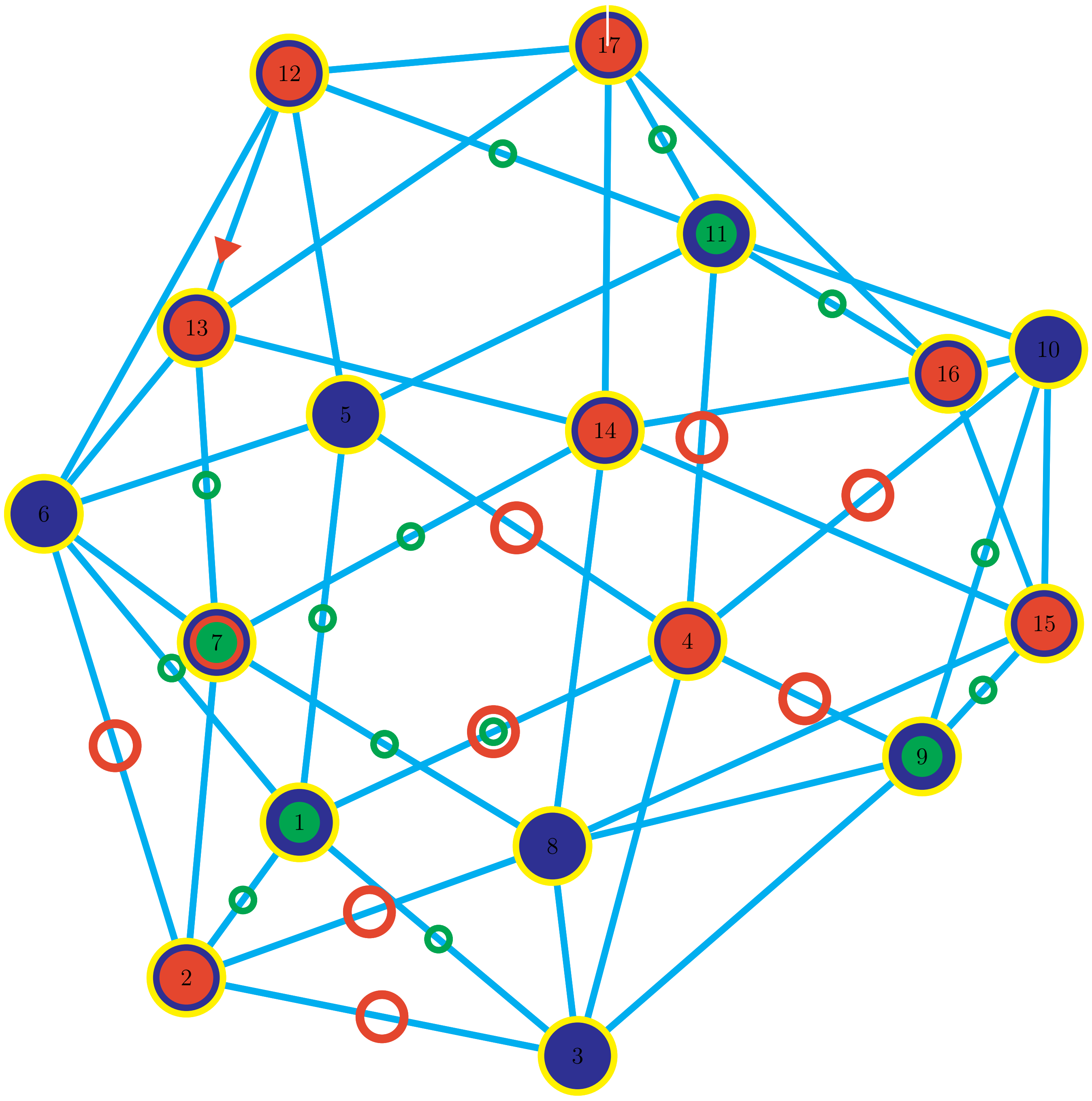

Figure 44. Instructions: 127: place edge 12→13 Red ArrowR,

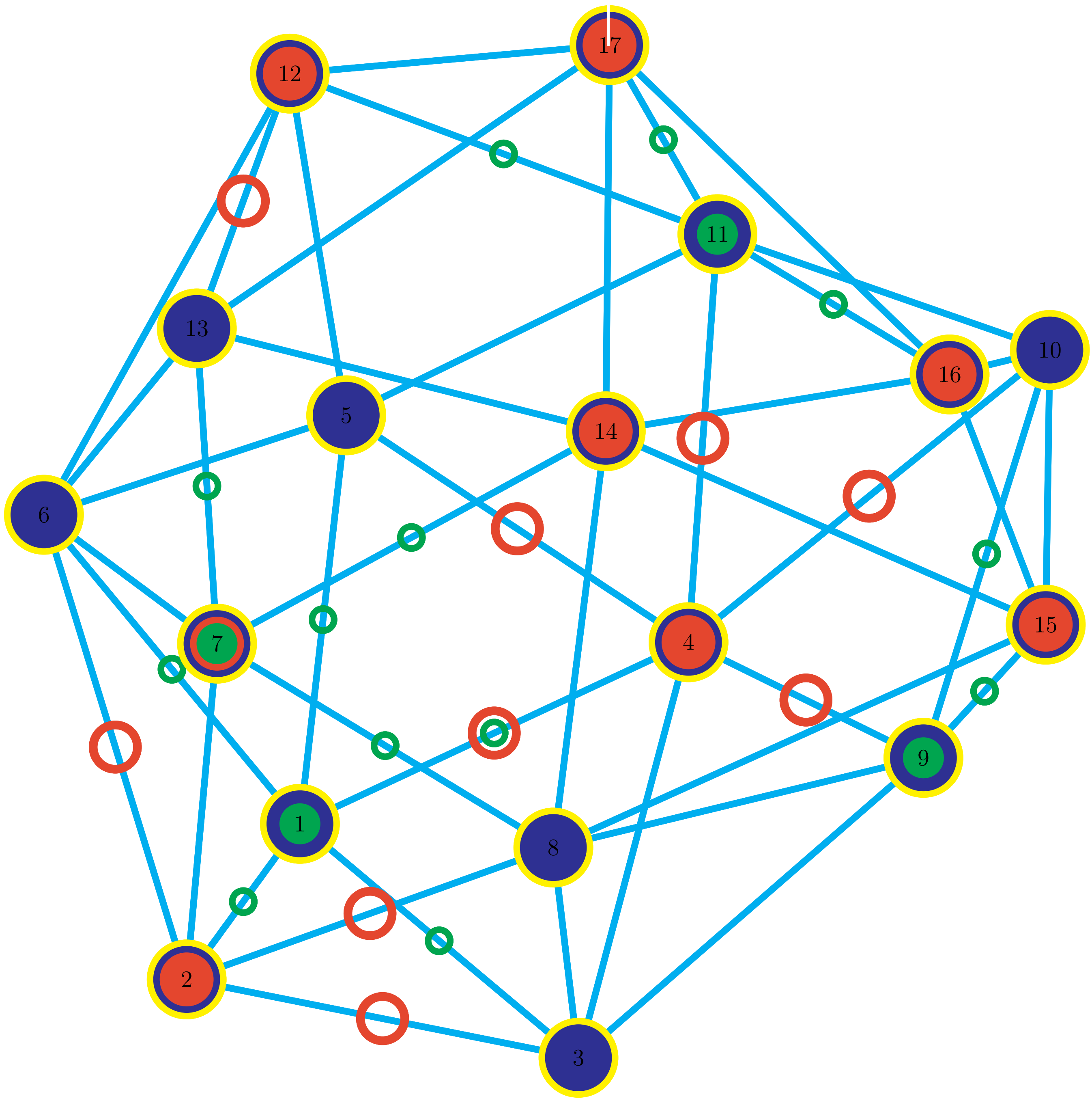

Figure 45. Instructions: 130: color edge 12–13 Red, 131: uncolor vertex 13 Red,

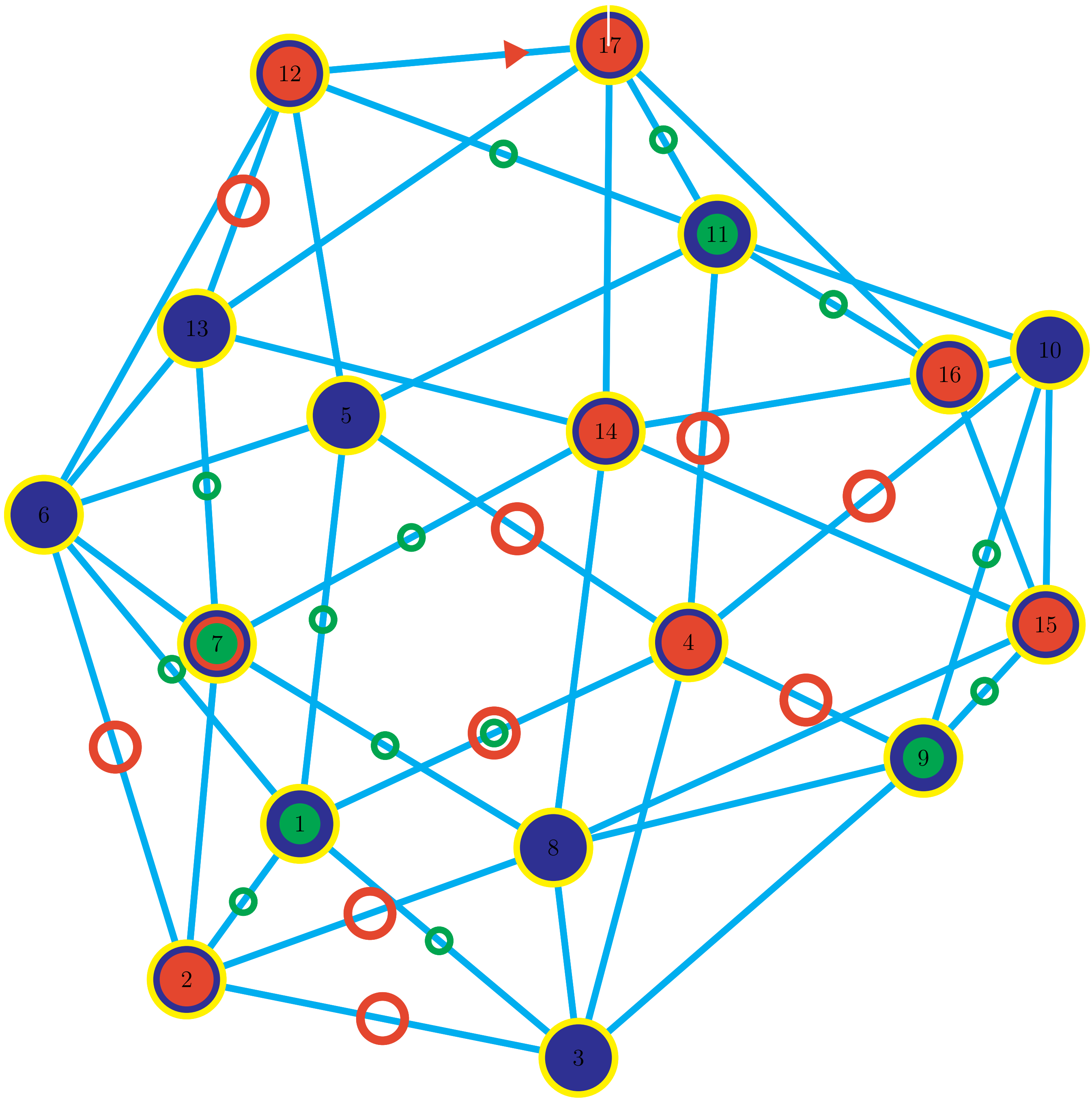

Figure 46. Instructions: 133: place edge 12→17 Red ArrowR,

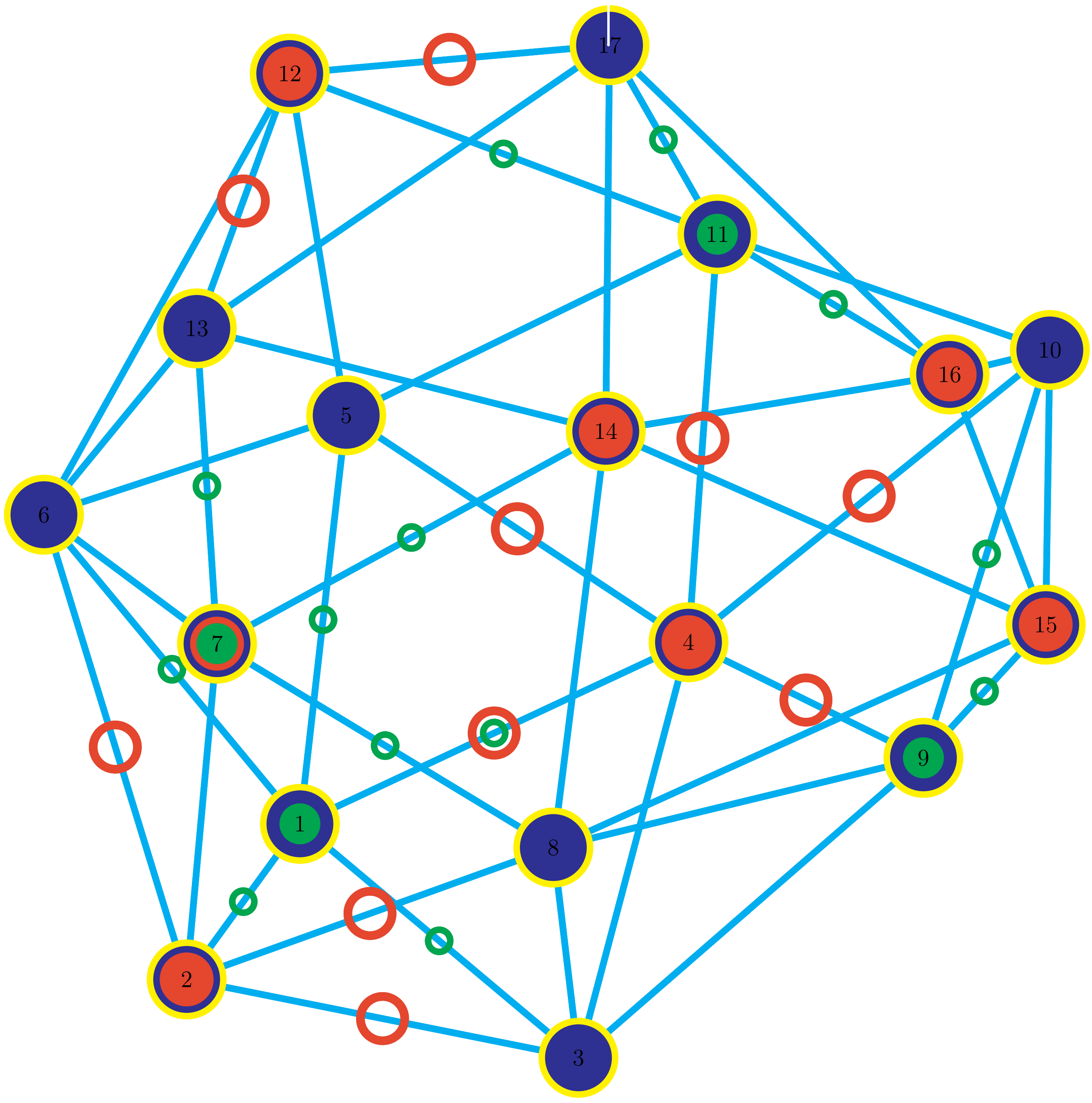

Figure 47. Instructions: 136: color edge 12–17 Red, 137: uncolor vertex 17 Red,

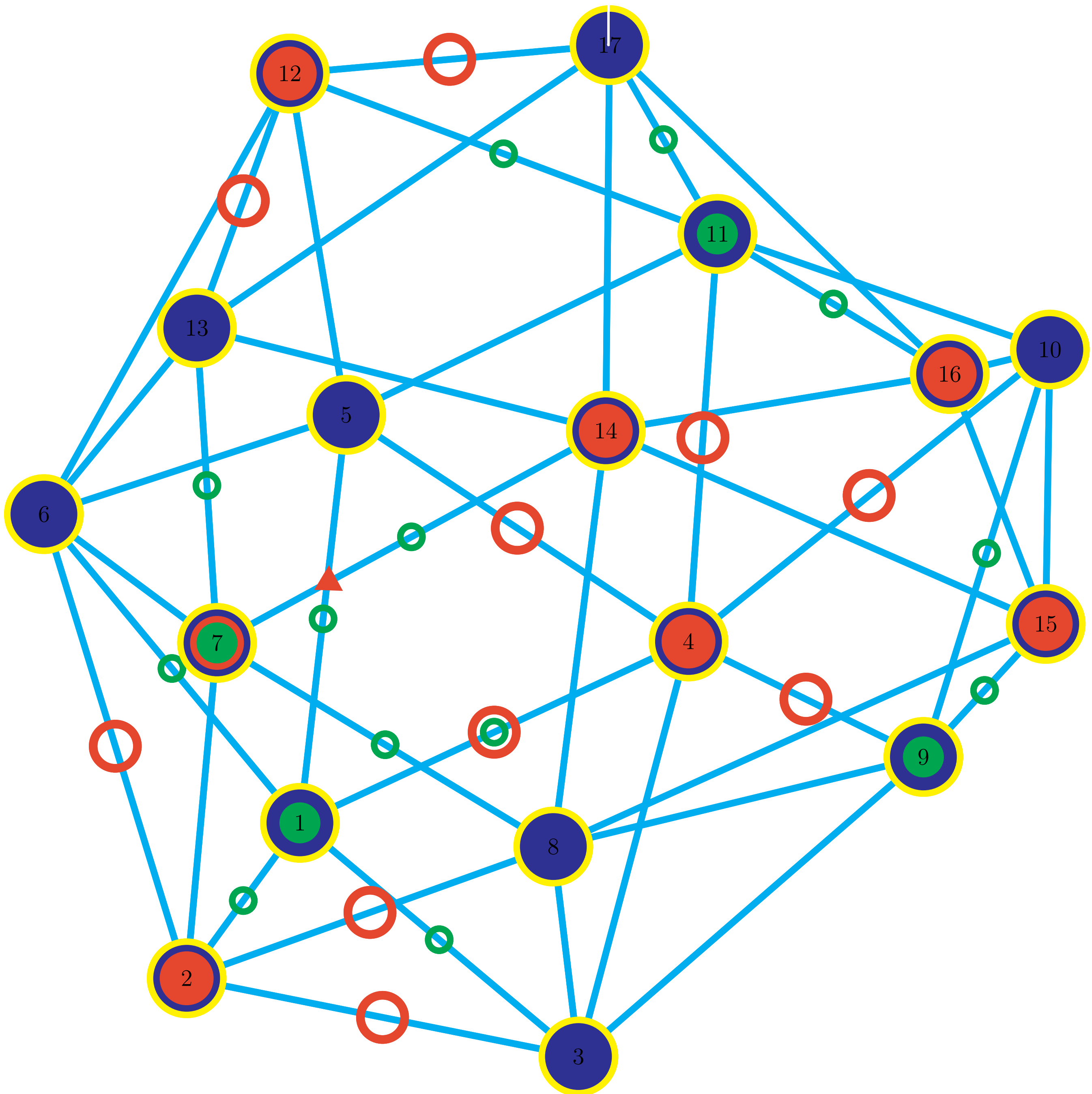

Figure 48. Instructions: 139: place edge 14→7 Red ArrowR,

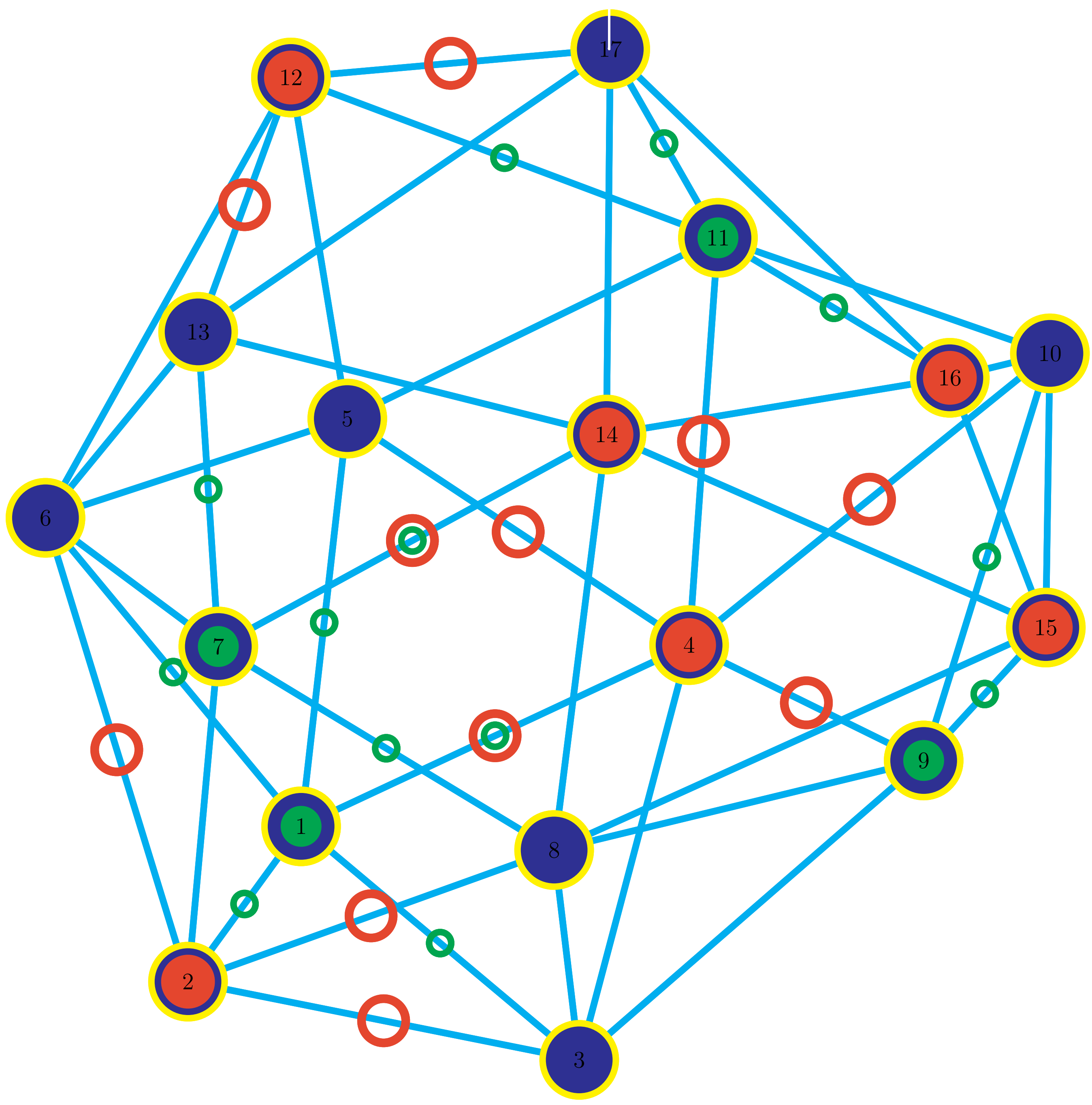

Figure 49. Instructions: 142: color edge 14–7 Red, 143: uncolor vertex 7 Red,

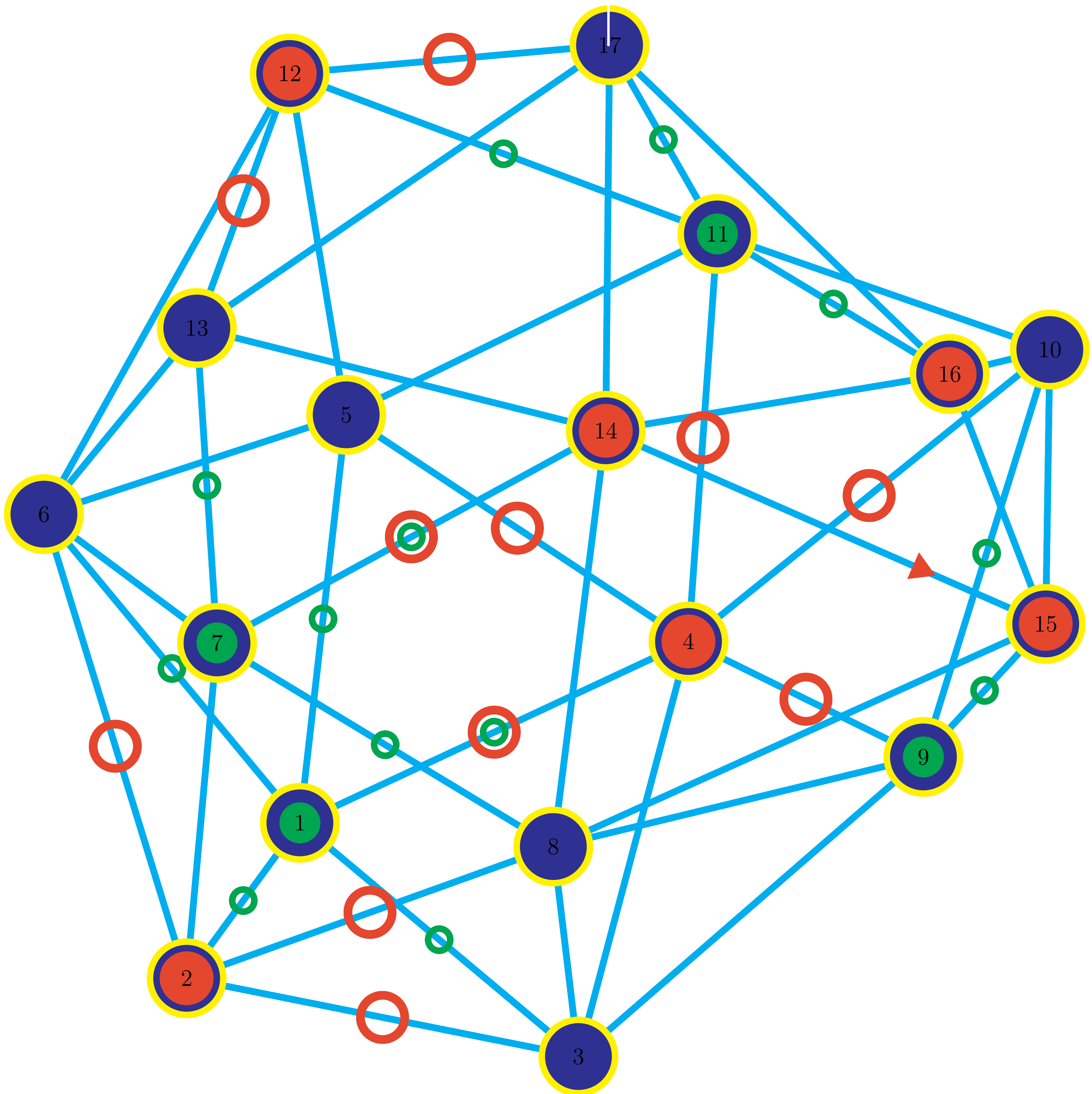

Figure 50. Instructions: 145: place edge 14→15 Red ArrowR,

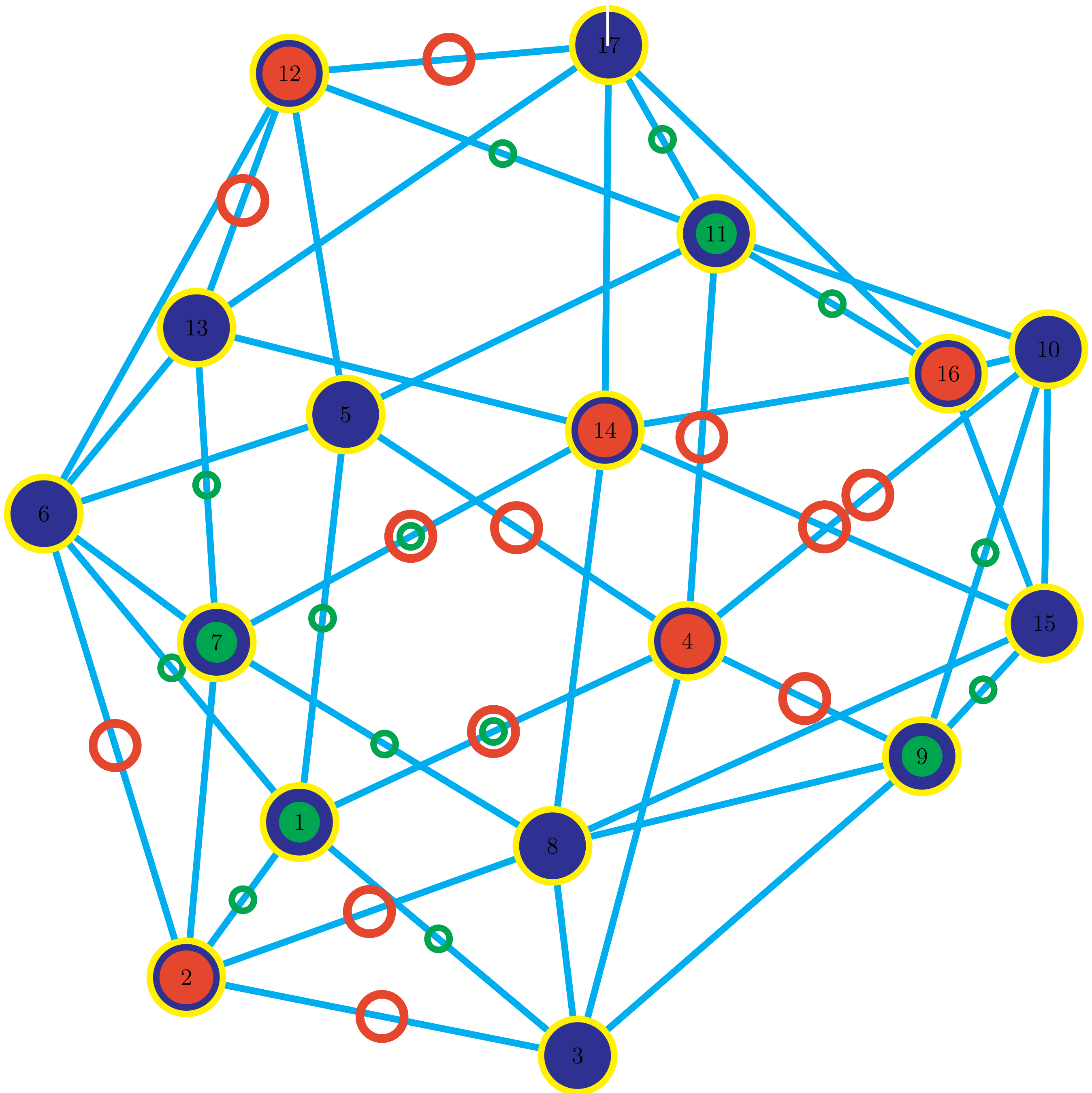

Figure 51. Instructions: 148: color edge 14–15 Red, 149: uncolor vertex 15 Red,

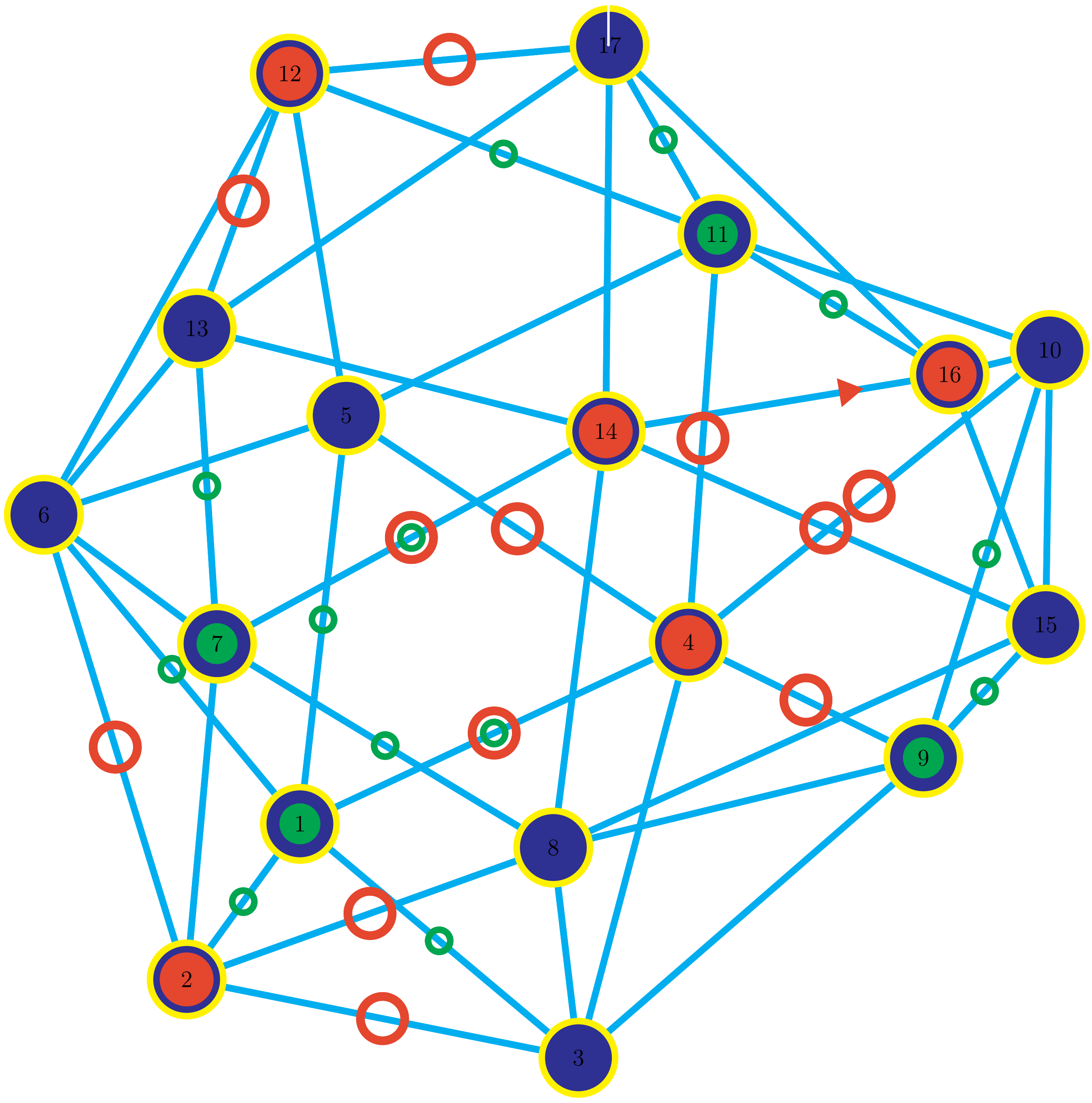

Figure 52. Instructions: 151: place edge 14→16 Red ArrowR,

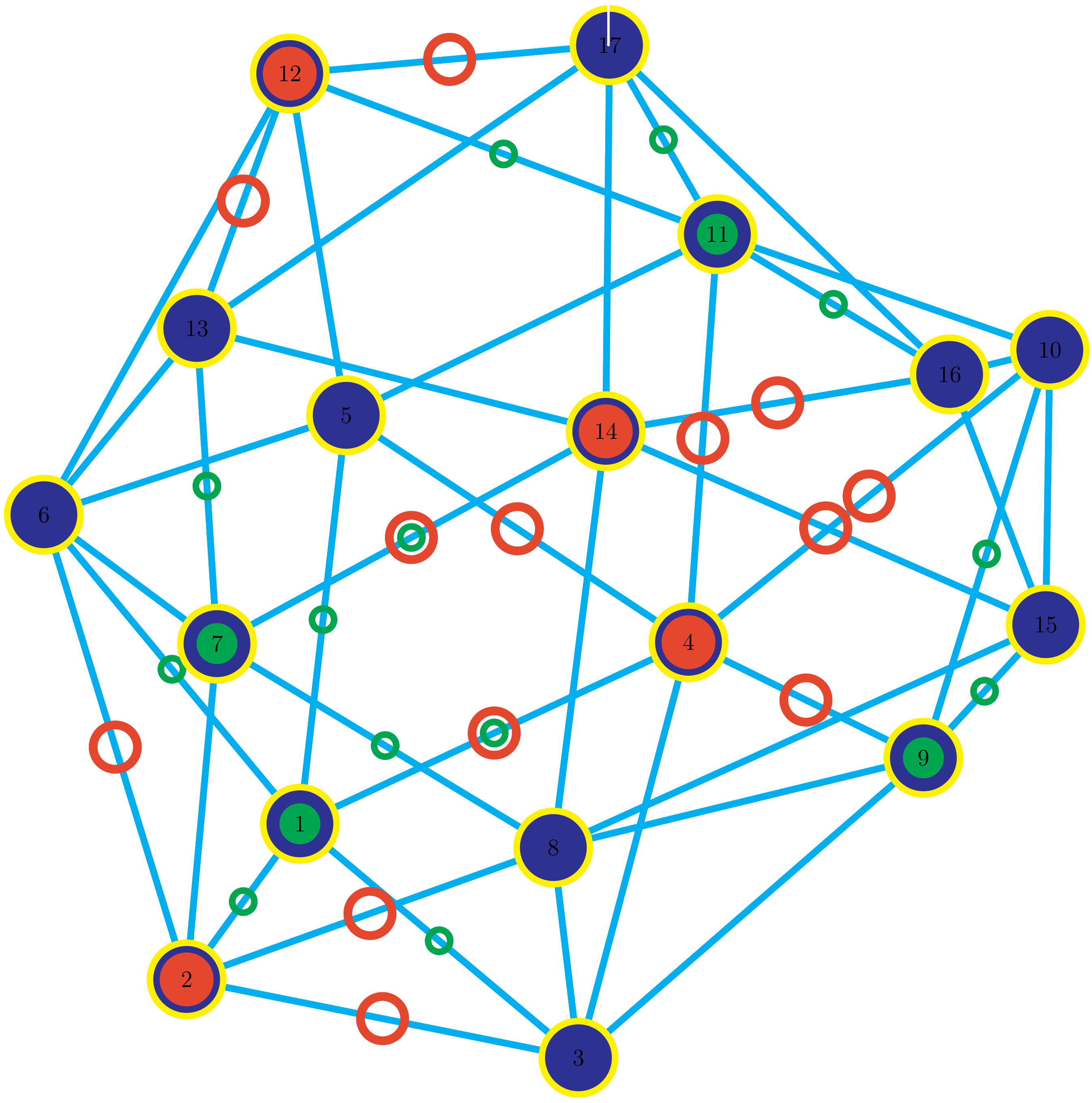

Figure 53. Instructions: 154: color edge 14–16 Red, 155: uncolor vertex 16 Red,

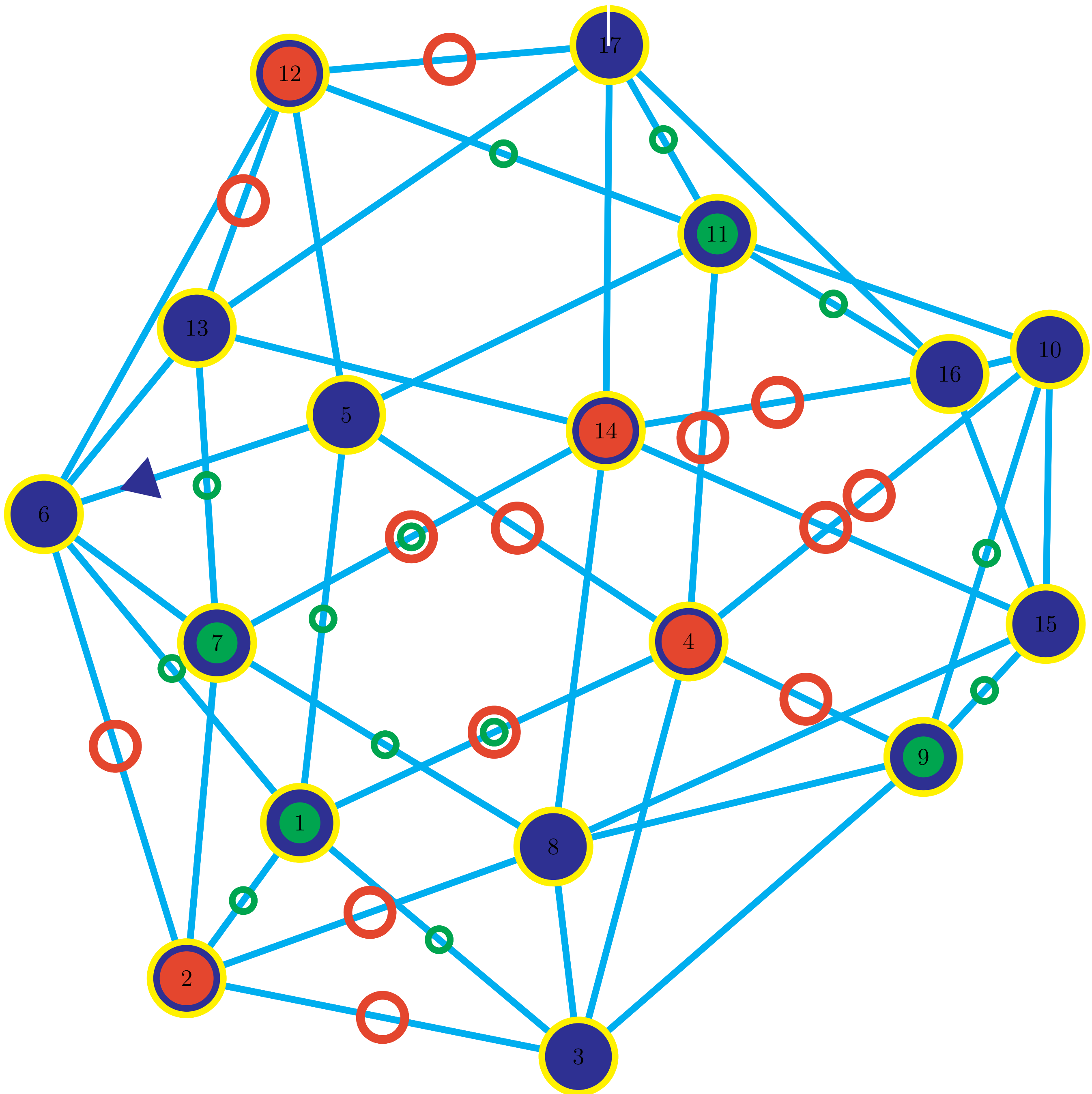

Figure 54. Instructions: 157: place edge 5→6 Blue ArrowR,

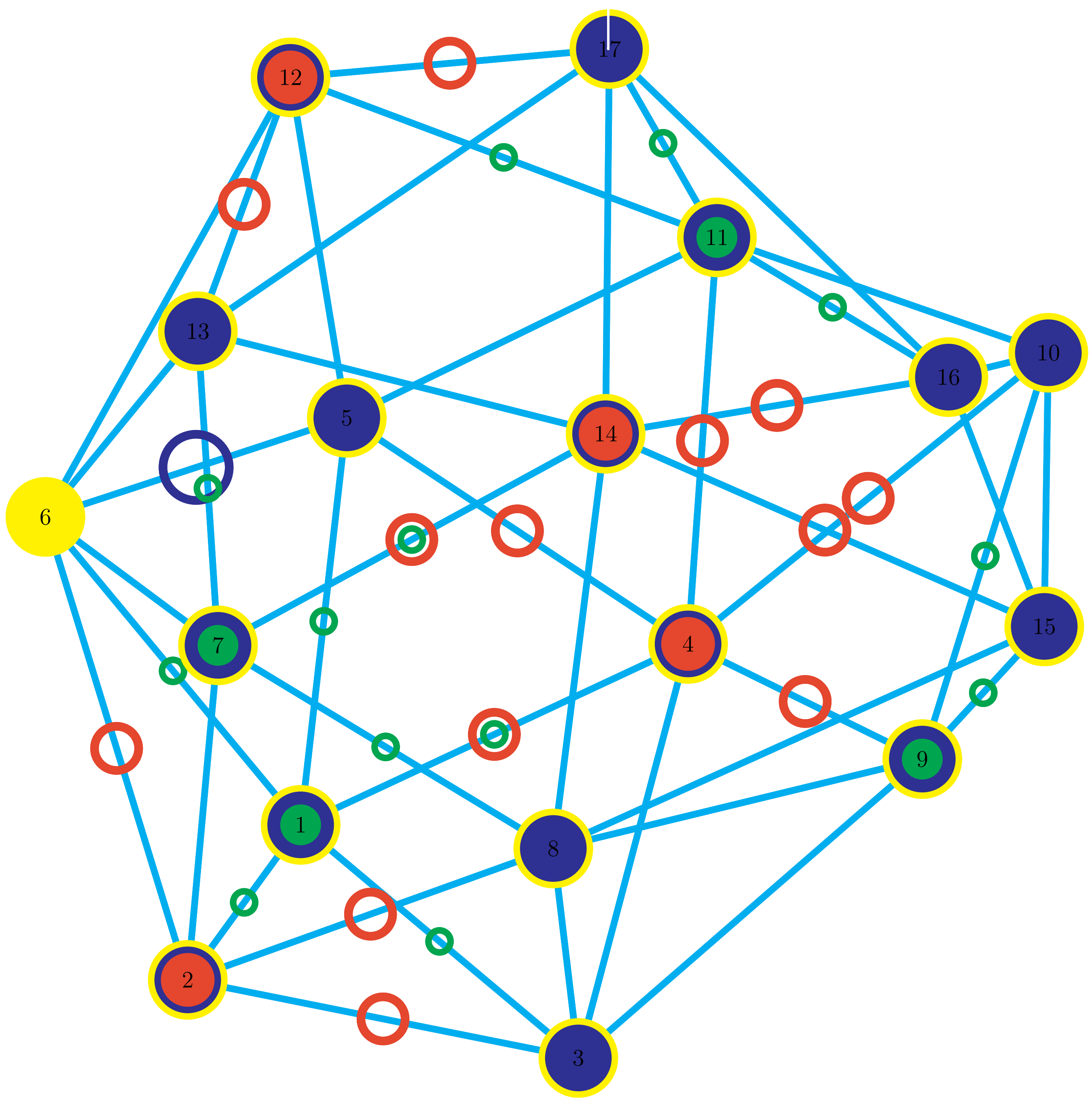

Figure 55. Instructions: 160: color edge 5–6 Blue, 161: uncolor vertex 6 Blue,

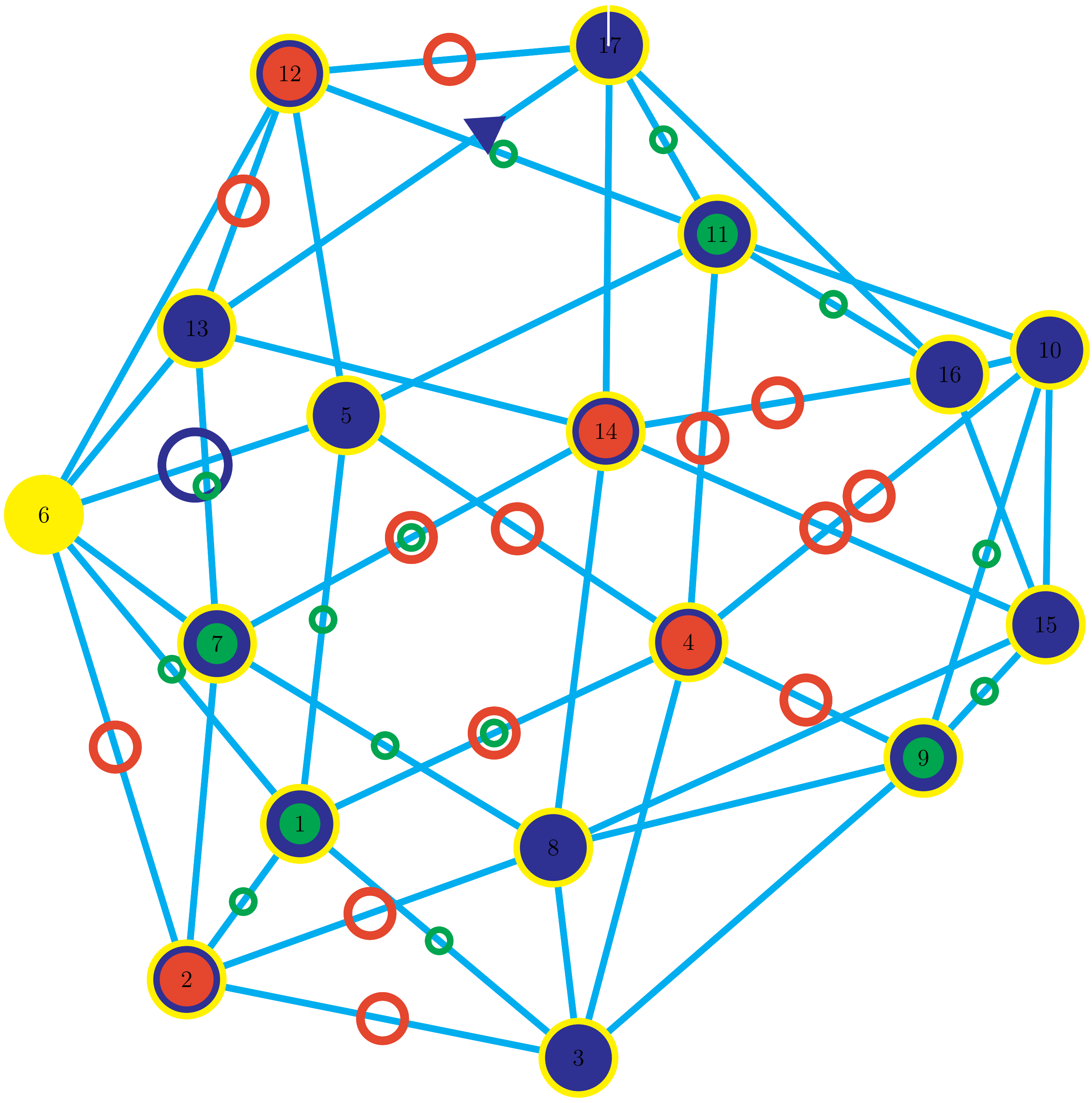

Figure 56. Instructions: 163: place edge 13→17 Blue ArrowR,

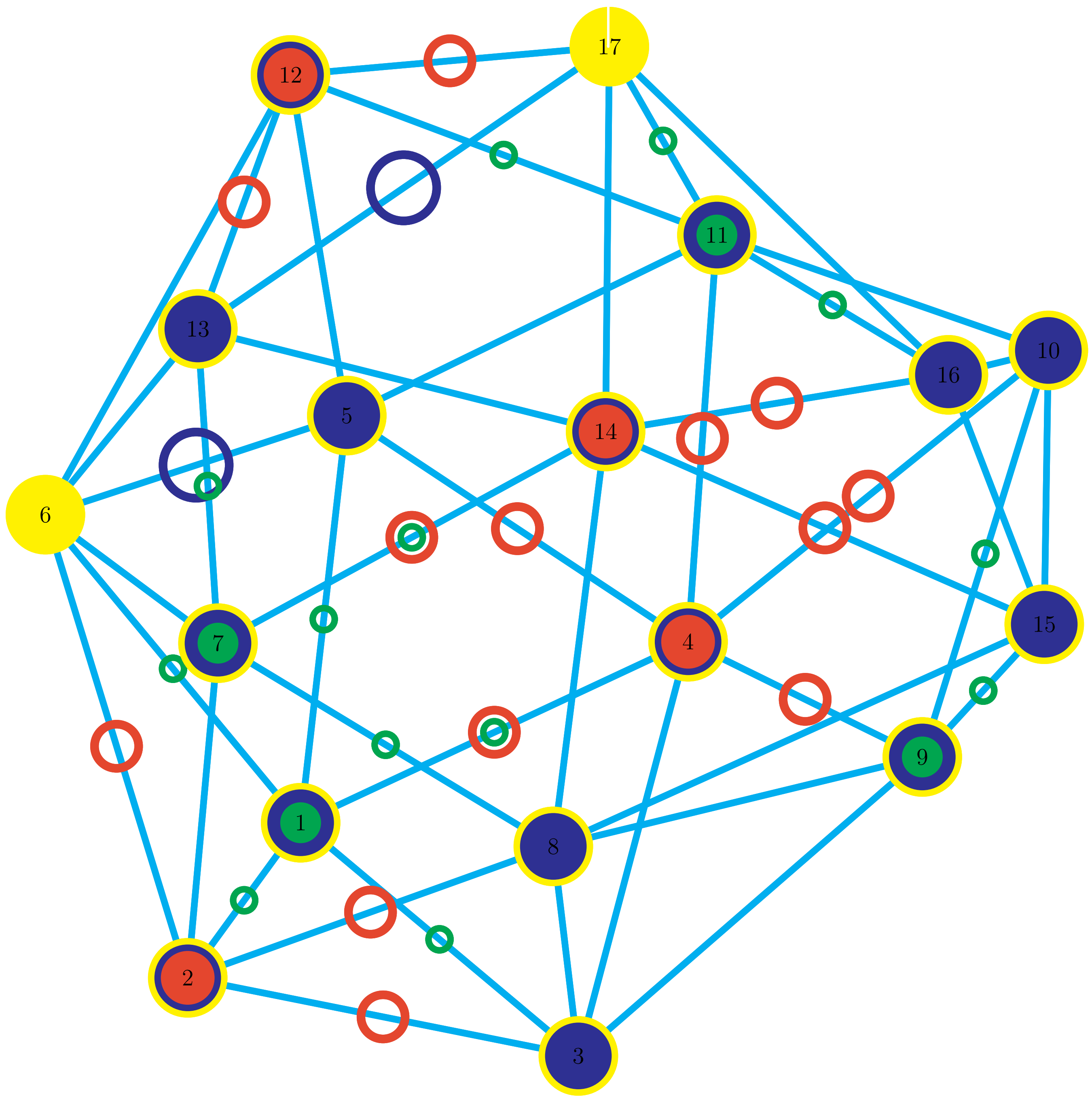

Figure 57. Instructions: 166: color edge 13–17 Blue, 167: uncolor vertex 17 Blue,

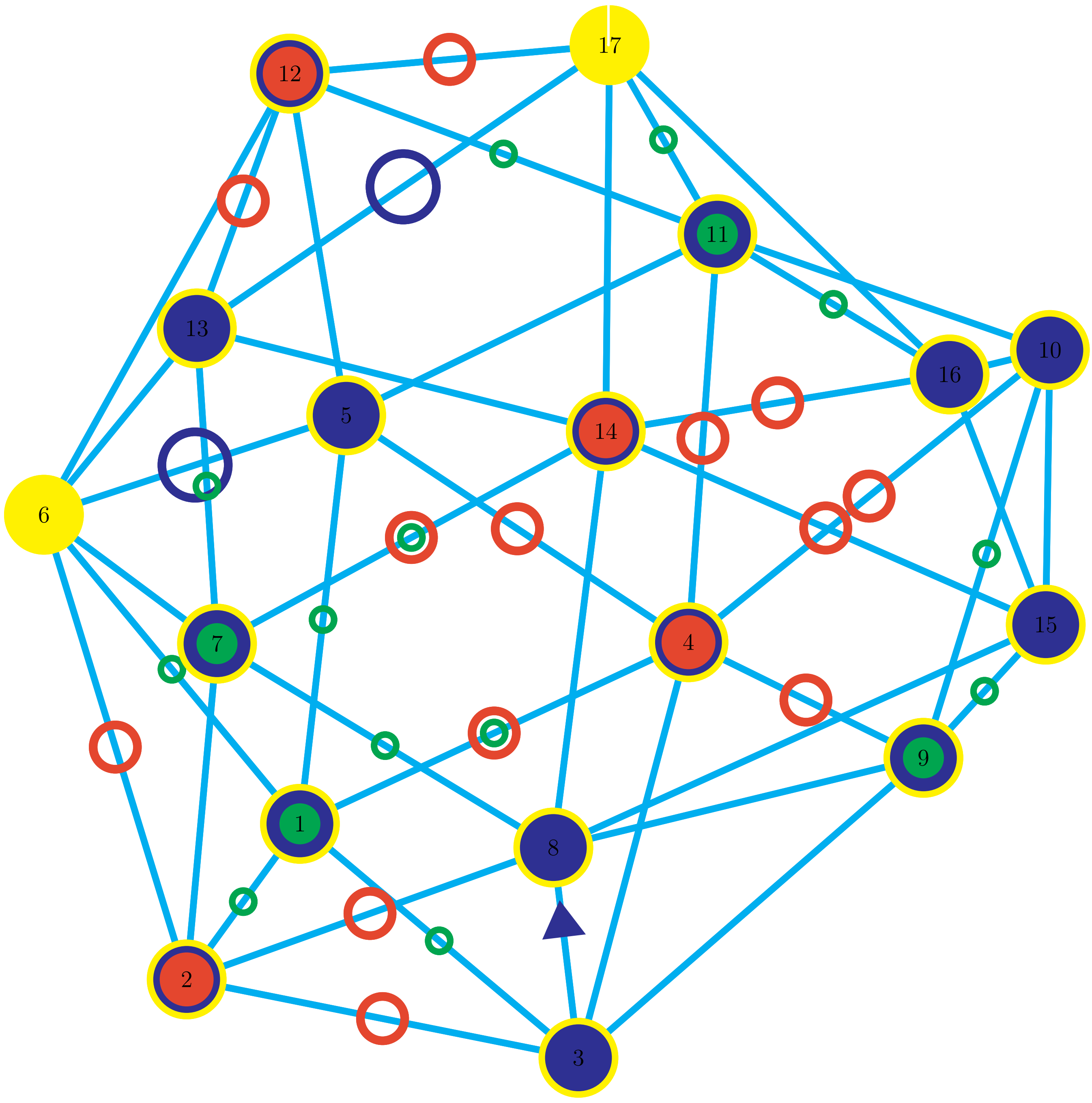

Figure 58. Instructions: 169: place edge 3→8 Blue ArrowR,

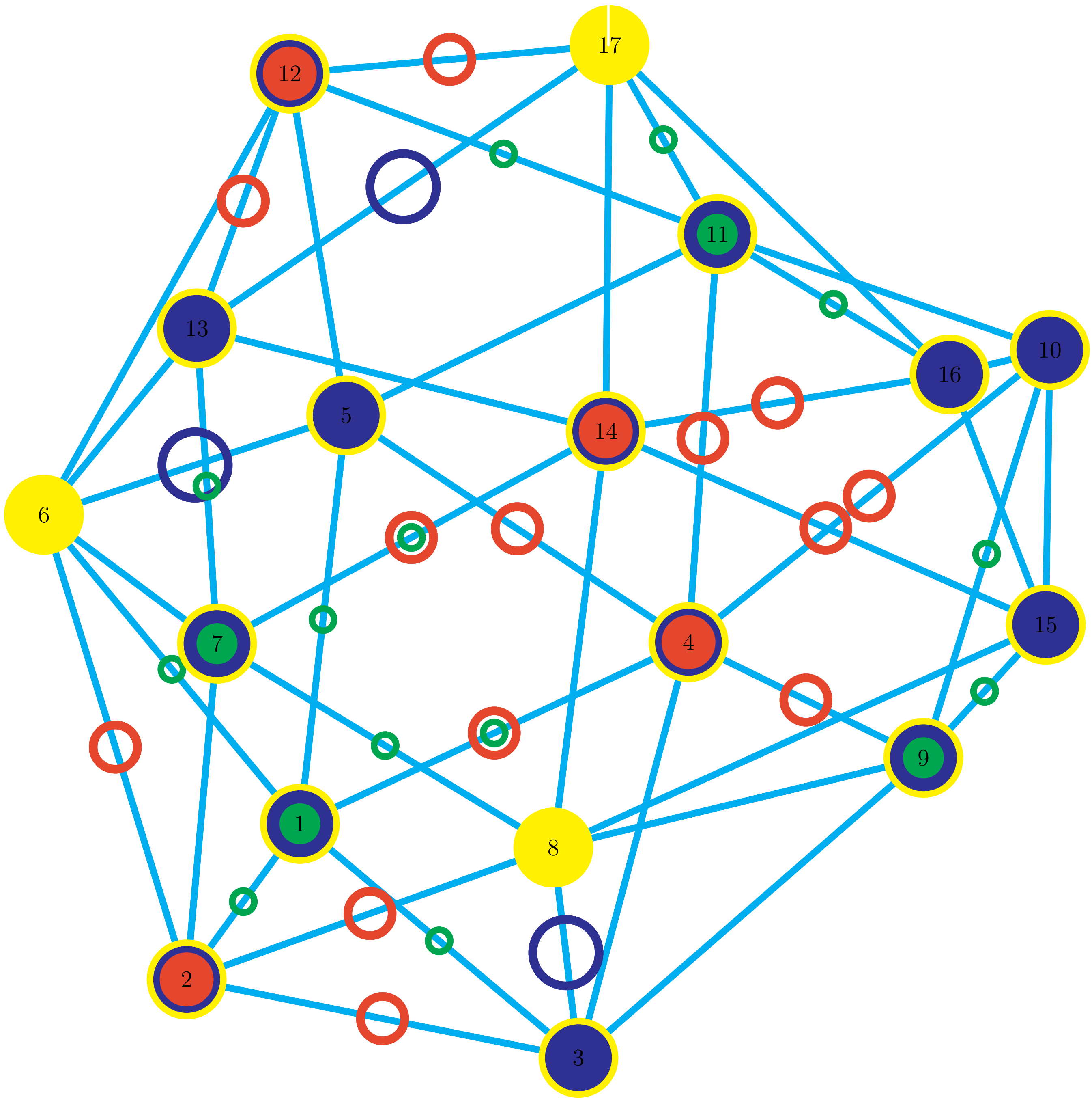

Figure 59. Instructions: 172: color edge 3–8 Blue, 173: uncolor vertex 8 Blue,

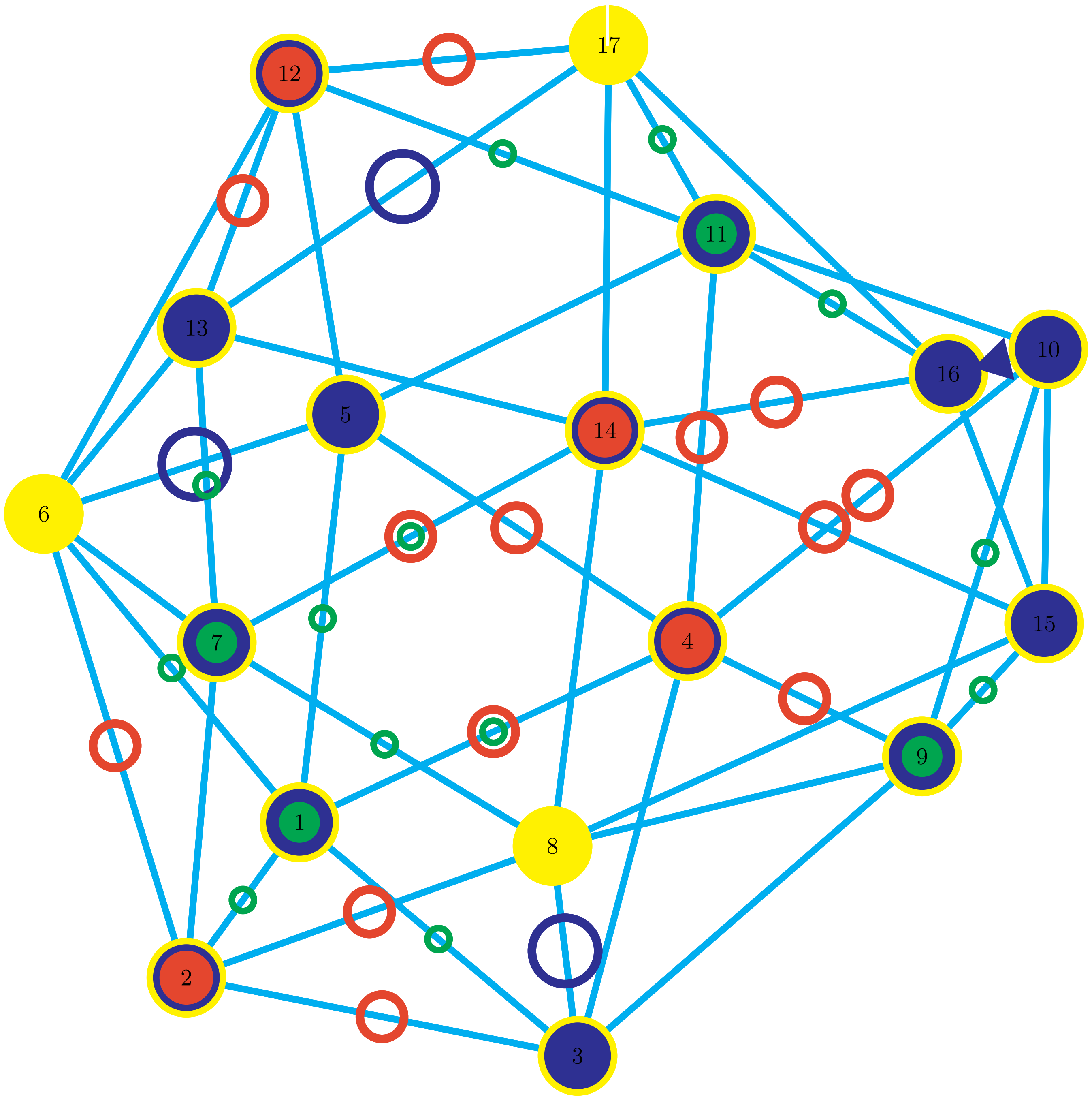

Figure 60. Instructions: 175: place edge 10→16 Blue ArrowR,

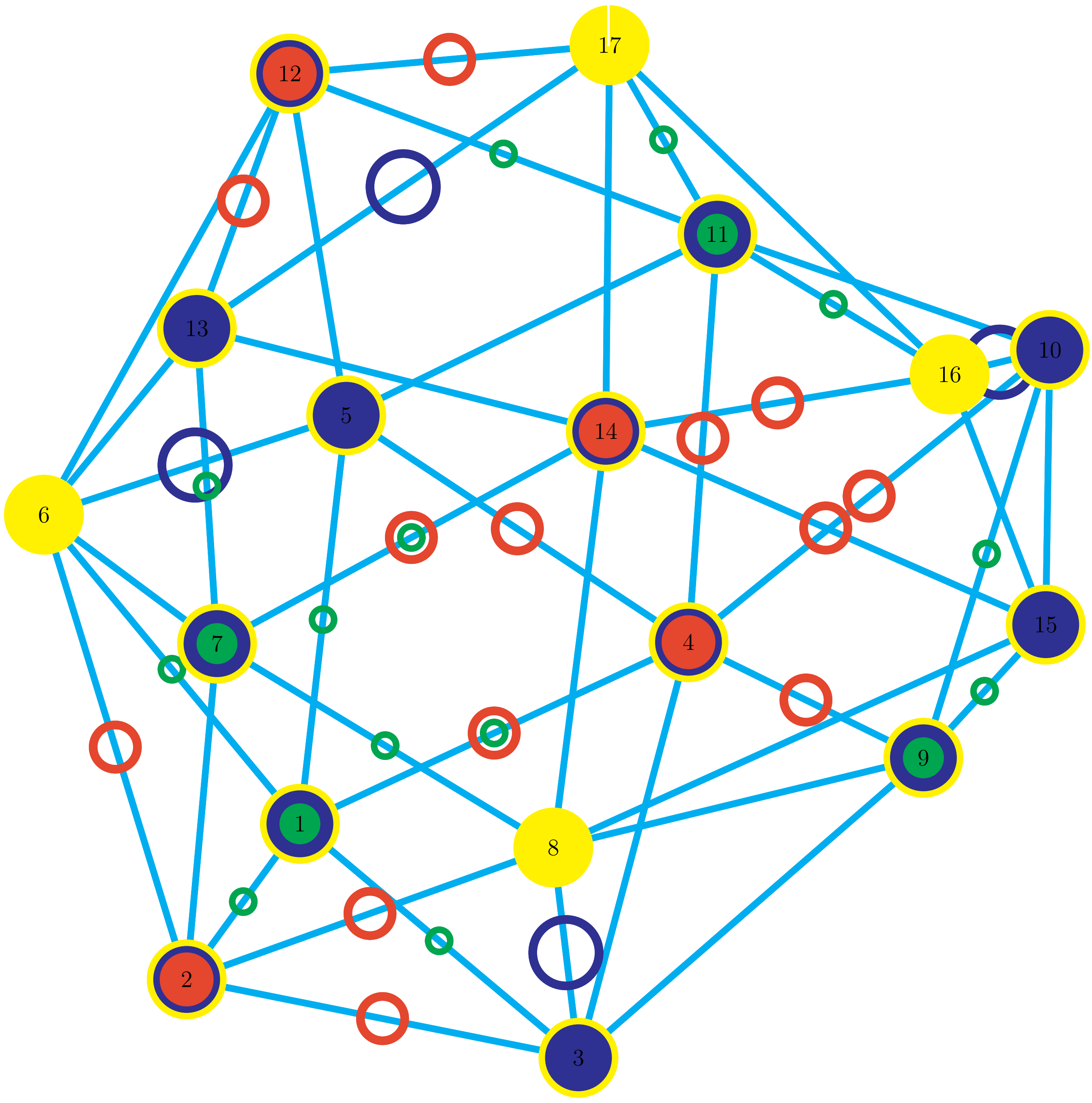

Figure 61. Instructions: 178: color edge 10–16 Blue, 179: uncolor vertex 16 Blue,

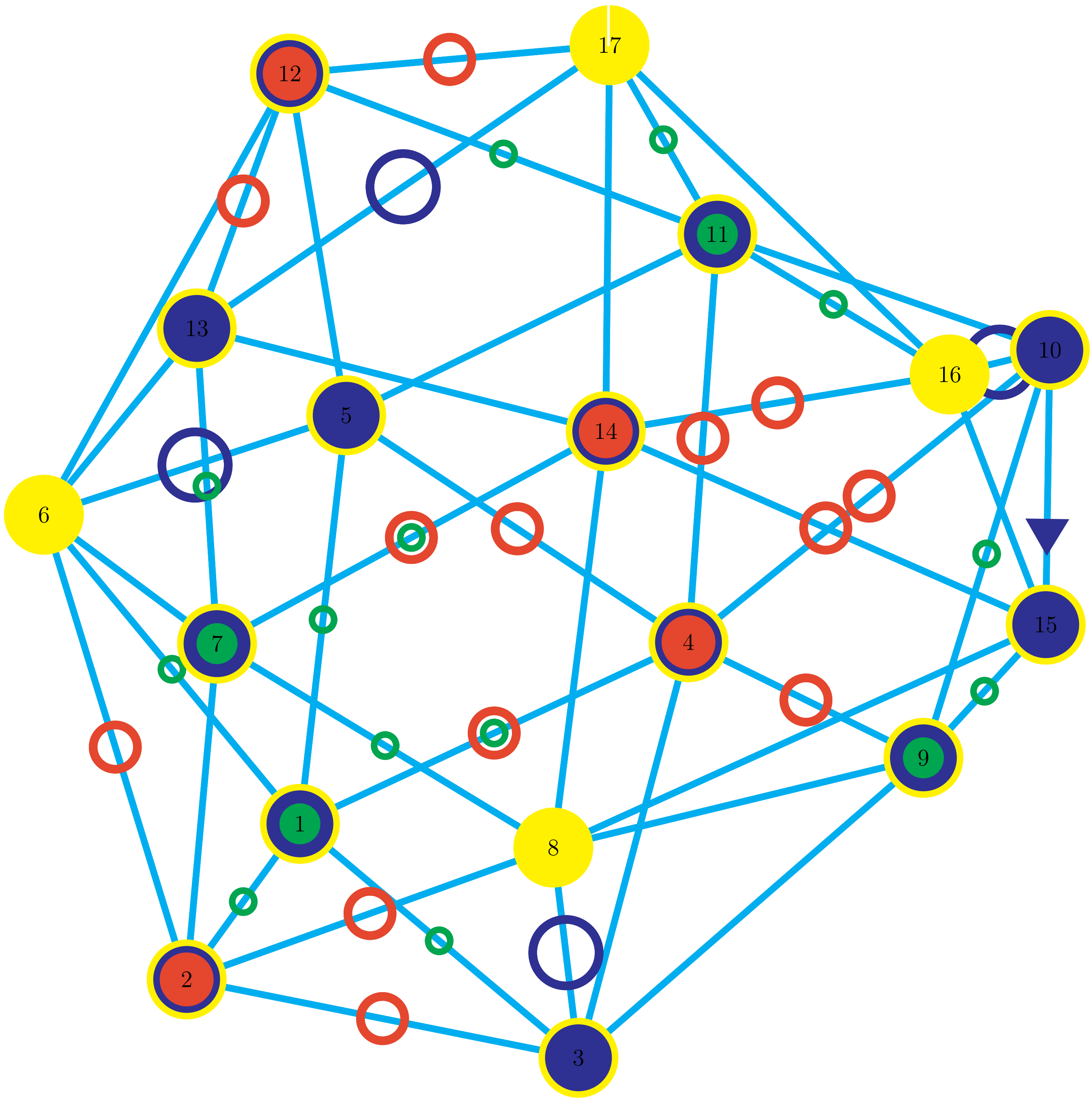

Figure 62. Instructions: 181: place edge 10→15 Blue ArrowR,

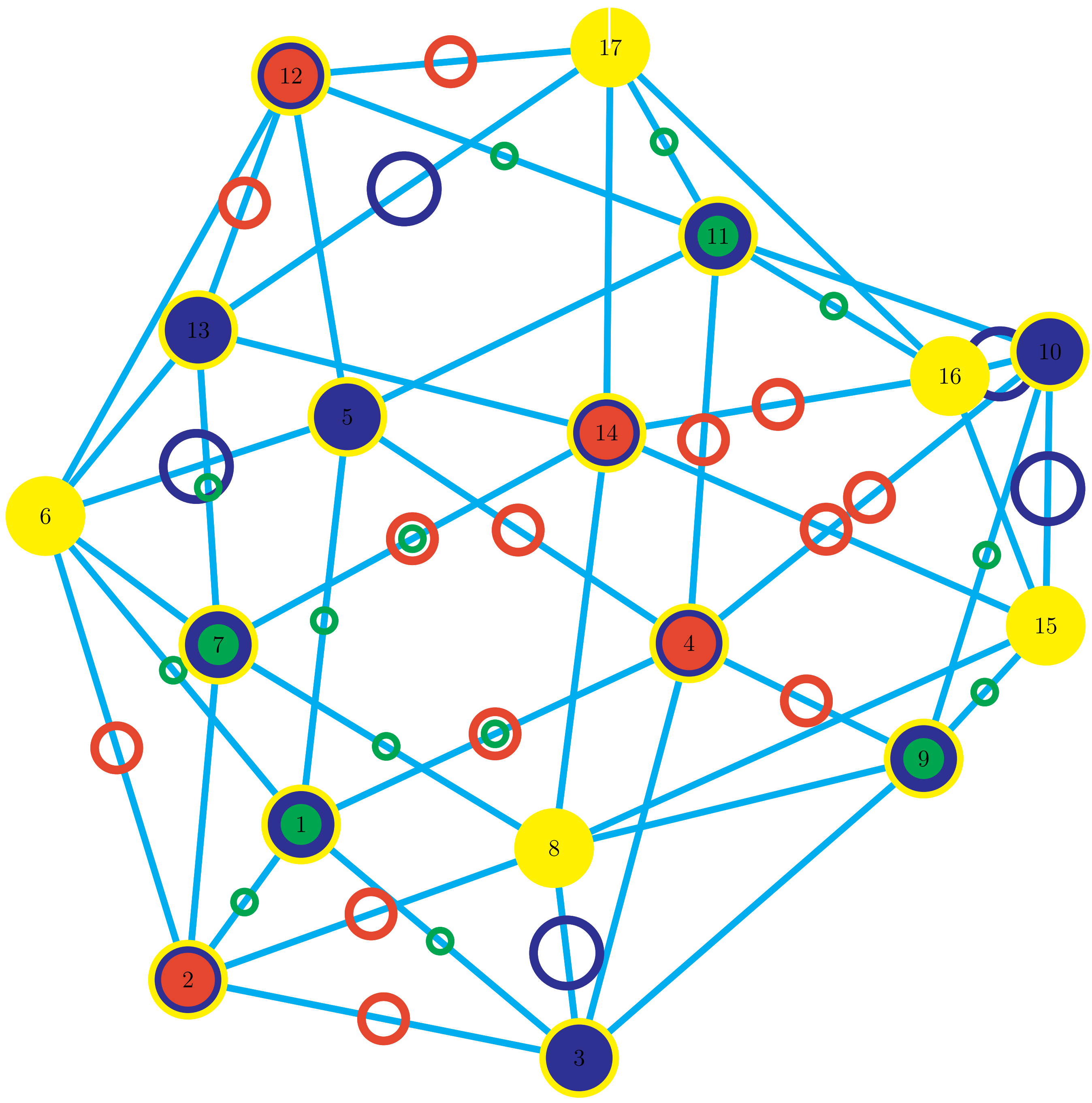

Figure 63. Instructions: 184: color edge 10–15 Blue, 185: uncolor vertex 15 Blue,

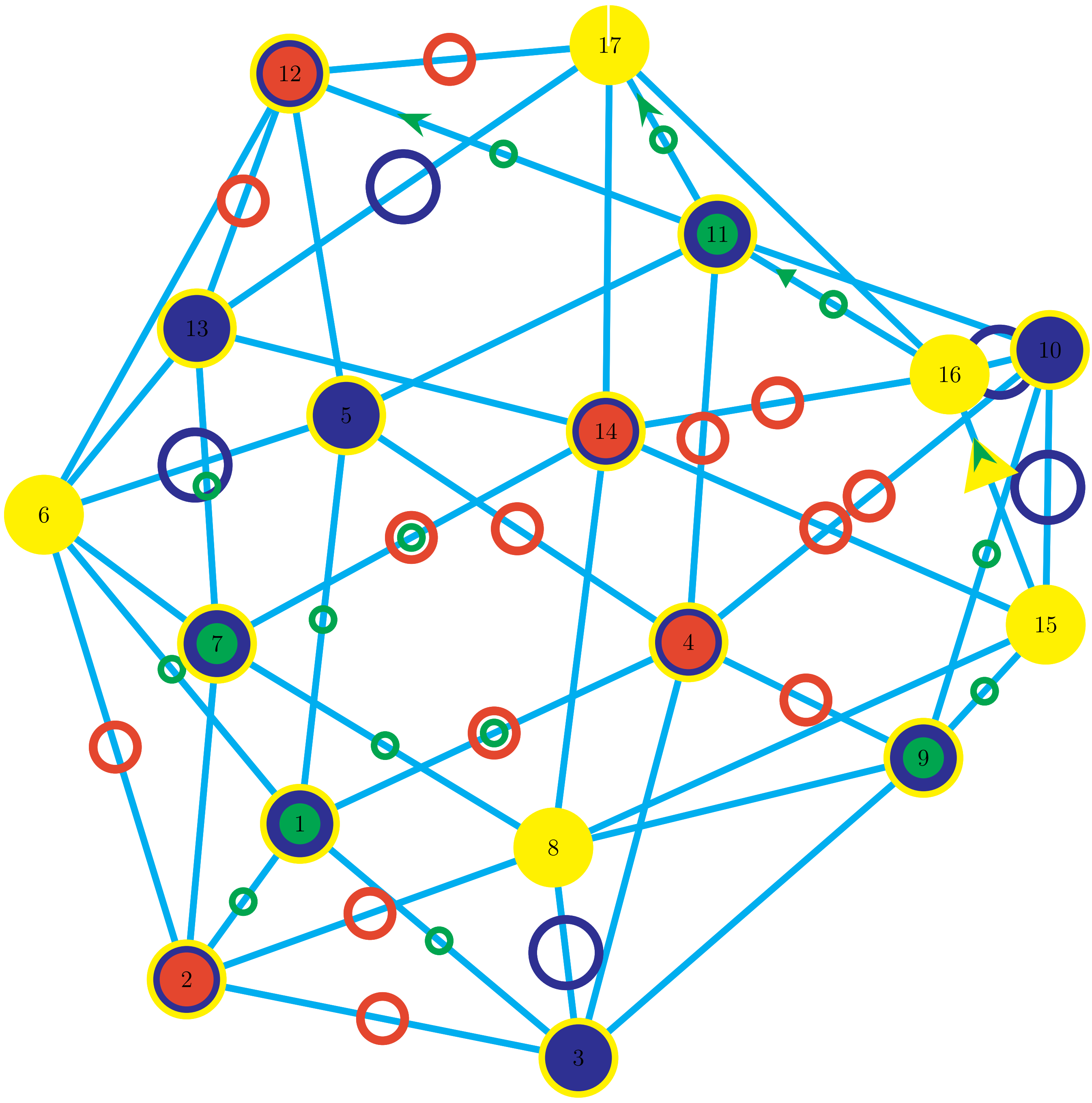

Figure 64. Instructions: 187: place edge 15→16 Yellow ArrowR, 188: place edge 15→16 Green ArrowI, 189: place edge 16→11 Green ArrowR, 190: place edge 11→17 Green ArrowI, 191: place edge 11→12 Green ArrowI,

This is the first alternating chain of length greater than 1. It is also branched. Insertion arrows are used for the first time. They are distinguished by having pointed tails whereas the deletion arrows have flat tails. There are two arrowheads pointing from vertex 15 to vertex 16. A yellow deletion arrow and a green insertion arrow. They indicate the deletion of a yellow checker on vertex 16 and the insertion of a green checker. There are also a deletion arrow pointing to vertex 11 and insertion arrows pointing to vertices 12 and 17.

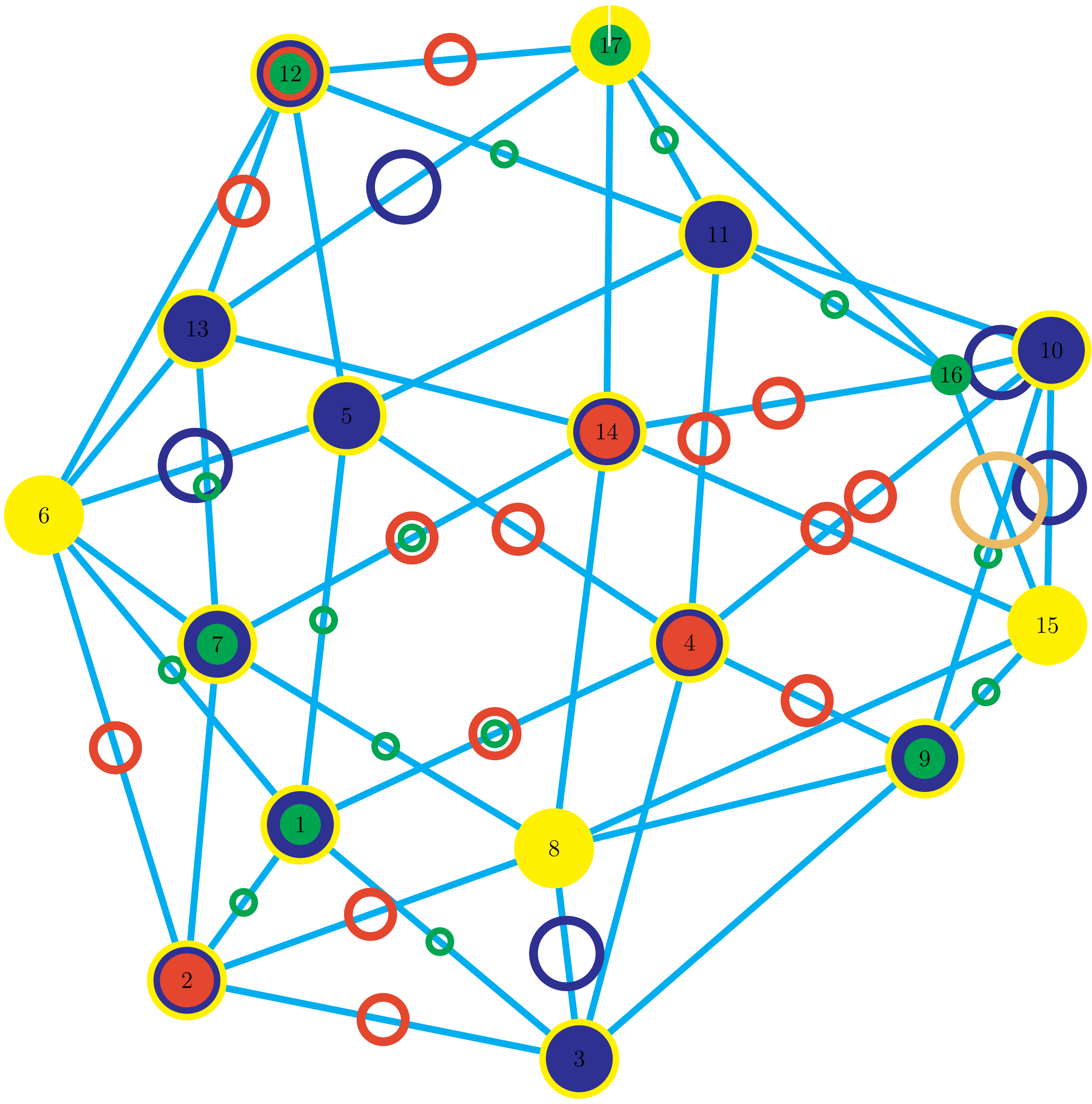

Figure 65. Instructions: 198: color edge 15–16 Yellow, 199: uncolor vertex 16 Yellow, 200: color vertex 16 Green, 201: uncolor vertex 11 Green, 202: color vertex 17 Green, 203: color vertex 12 Green,

The checkers have been moved according to the arrows and a new Yellow circle has been placed on edge 15-16 indicating that there is no conflict on this edge wrt Yellow. Also Properties A and B obtain.

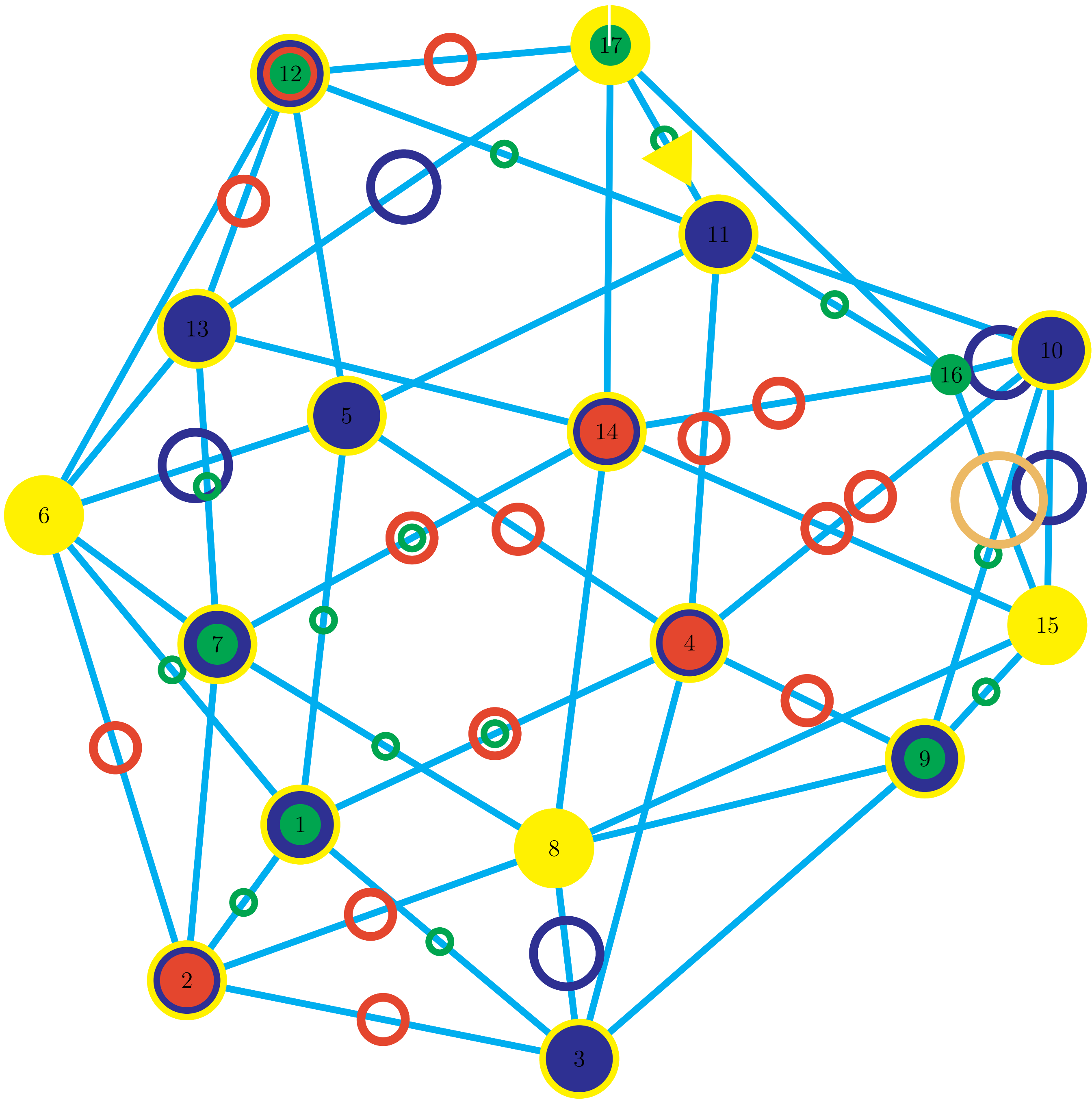

Figure 66. Instructions: 205: place edge 17→11 Yellow ArrowR,

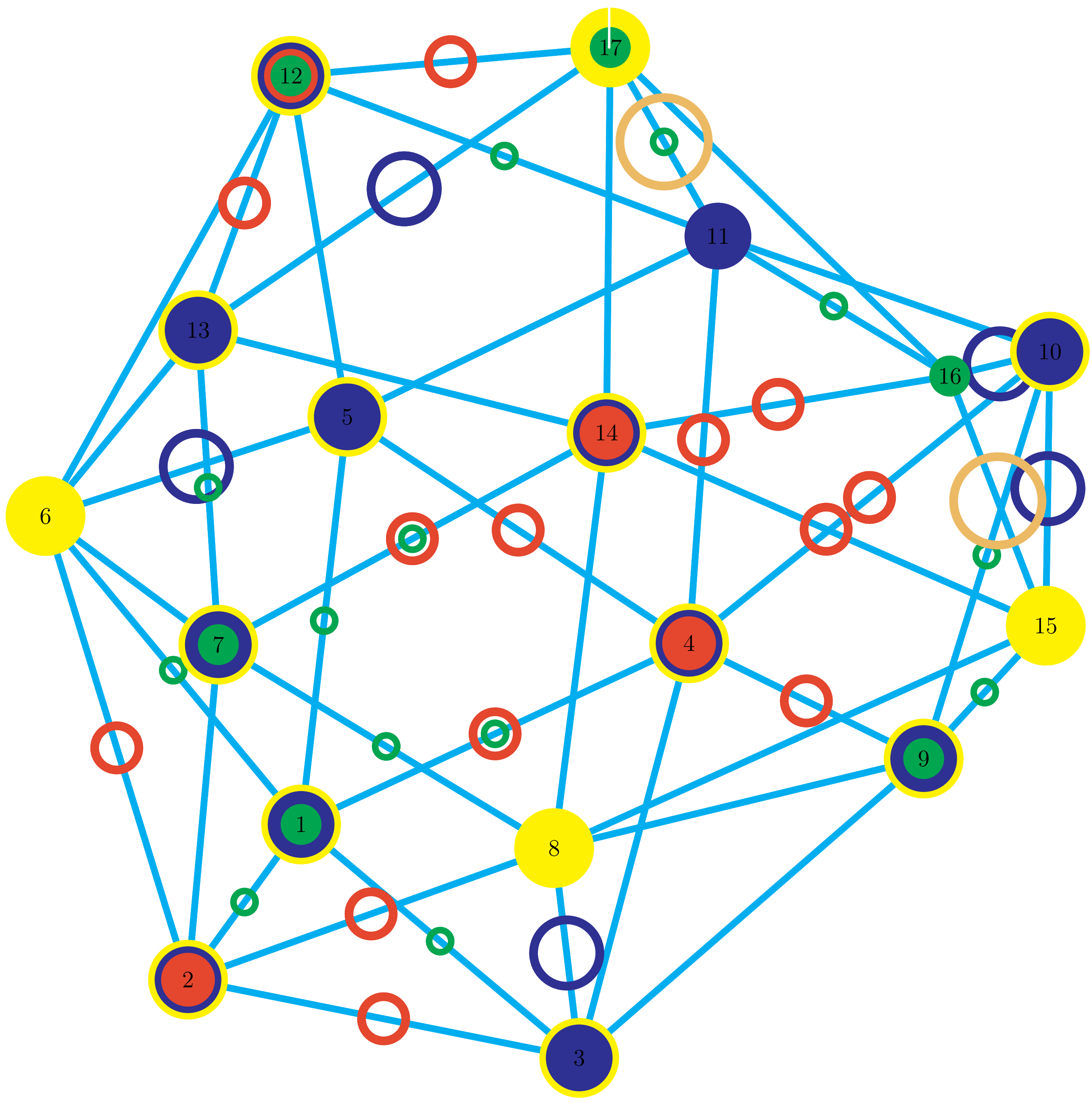

Figure 67. Instructions: 208: color edge 11–17 Yellow, 209: uncolor vertex 11 Yellow,

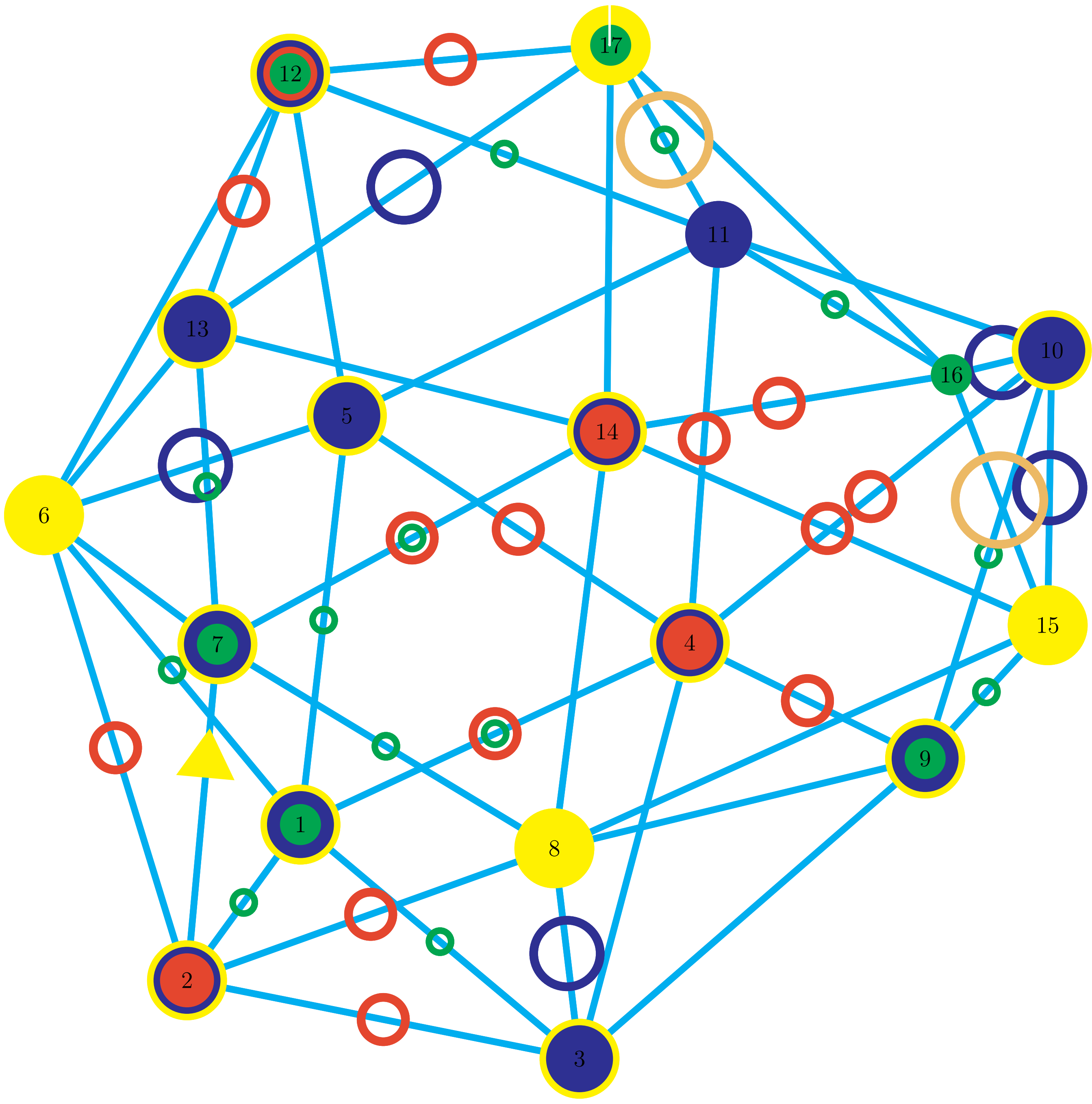

Figure 68. Instructions: 211: place edge 2→7 Yellow ArrowR,

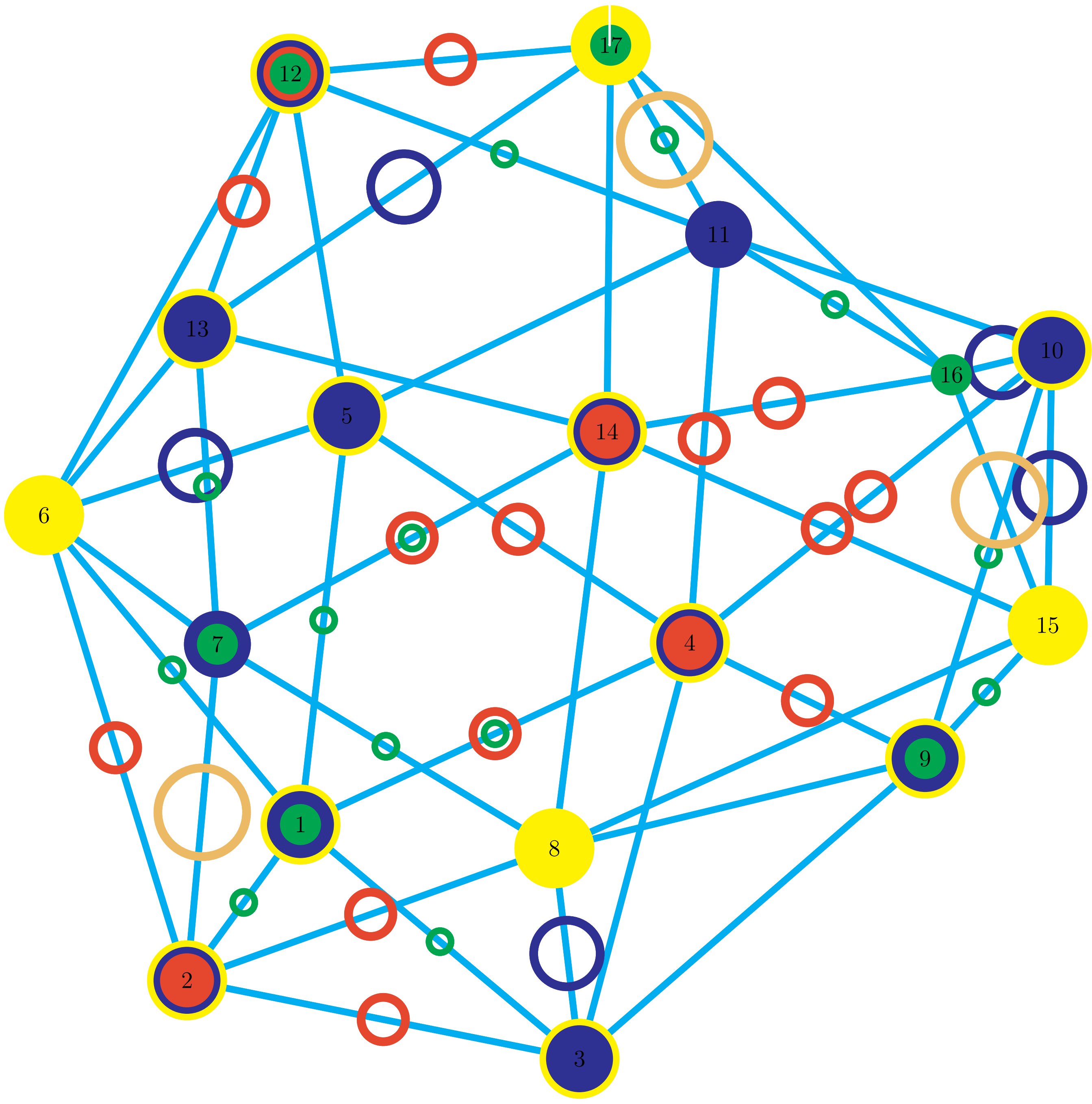

Figure 69. Instructions: 214: color edge 2–7 Yellow, 215: uncolor vertex 7 Yellow,

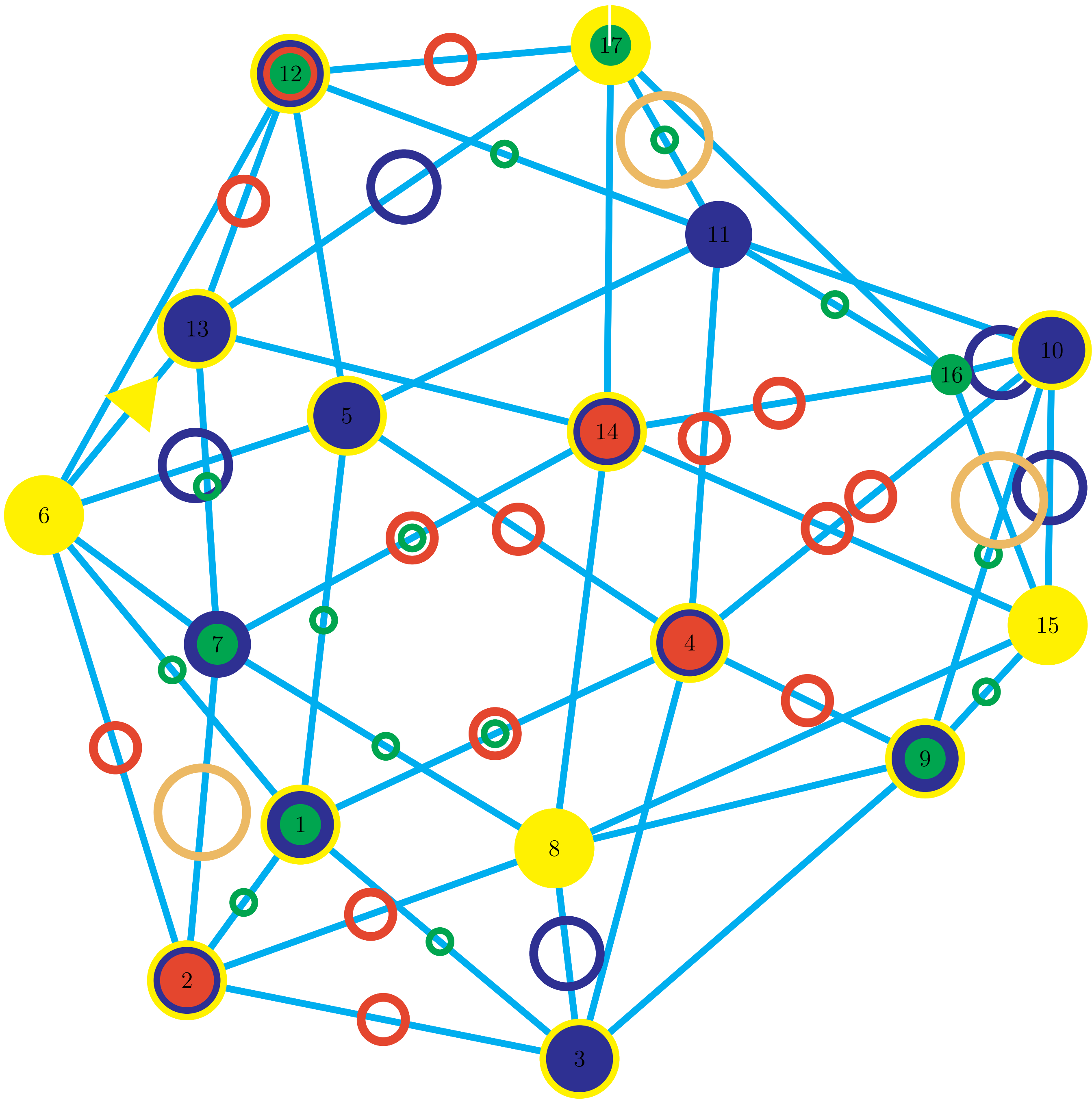

Figure 70. Instructions: 217: place edge 6→13 Yellow ArrowR,

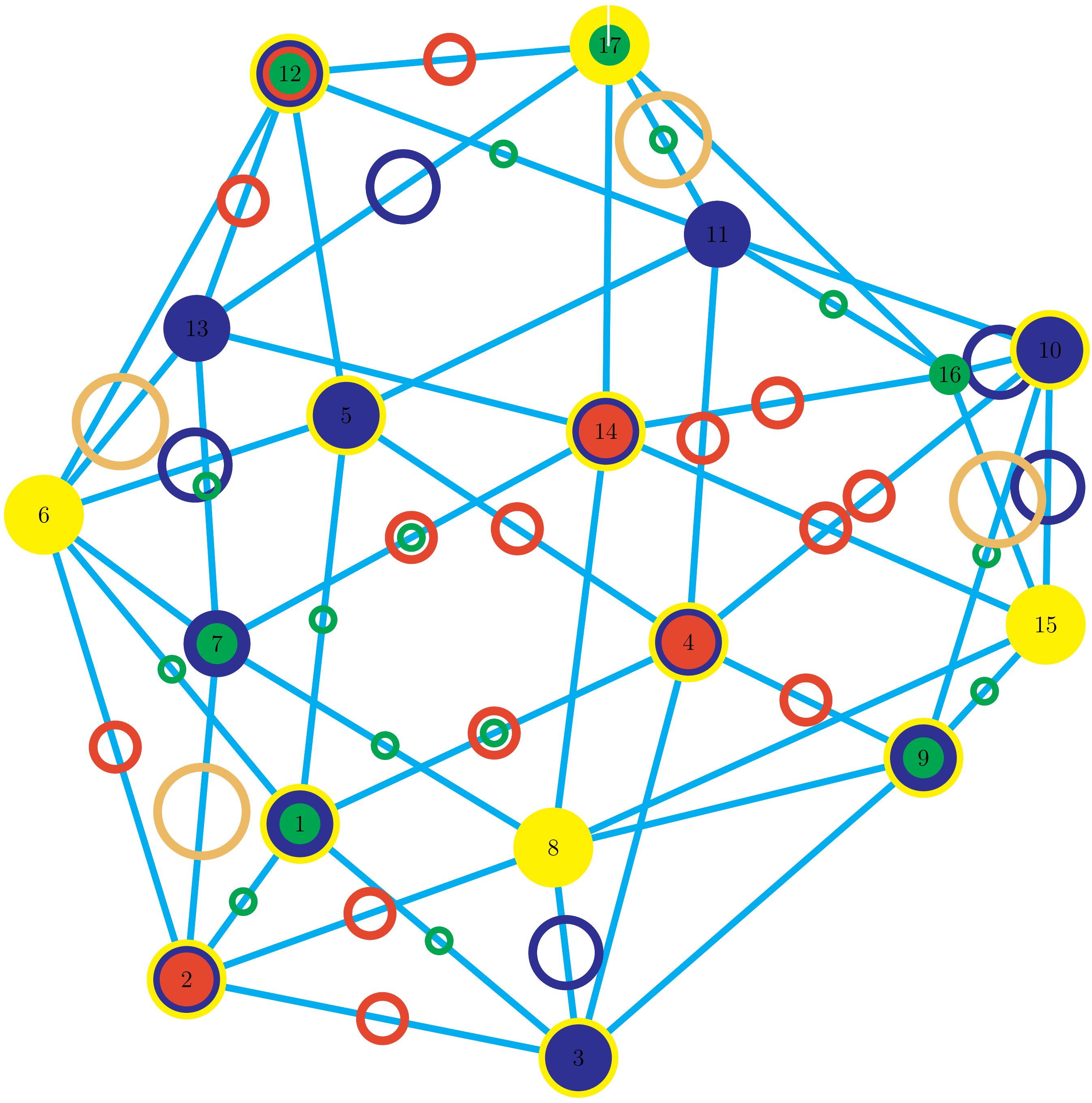

Figure 71. Instructions: 220: color edge 6–13 Yellow, 221: uncolor vertex 13 Yellow,

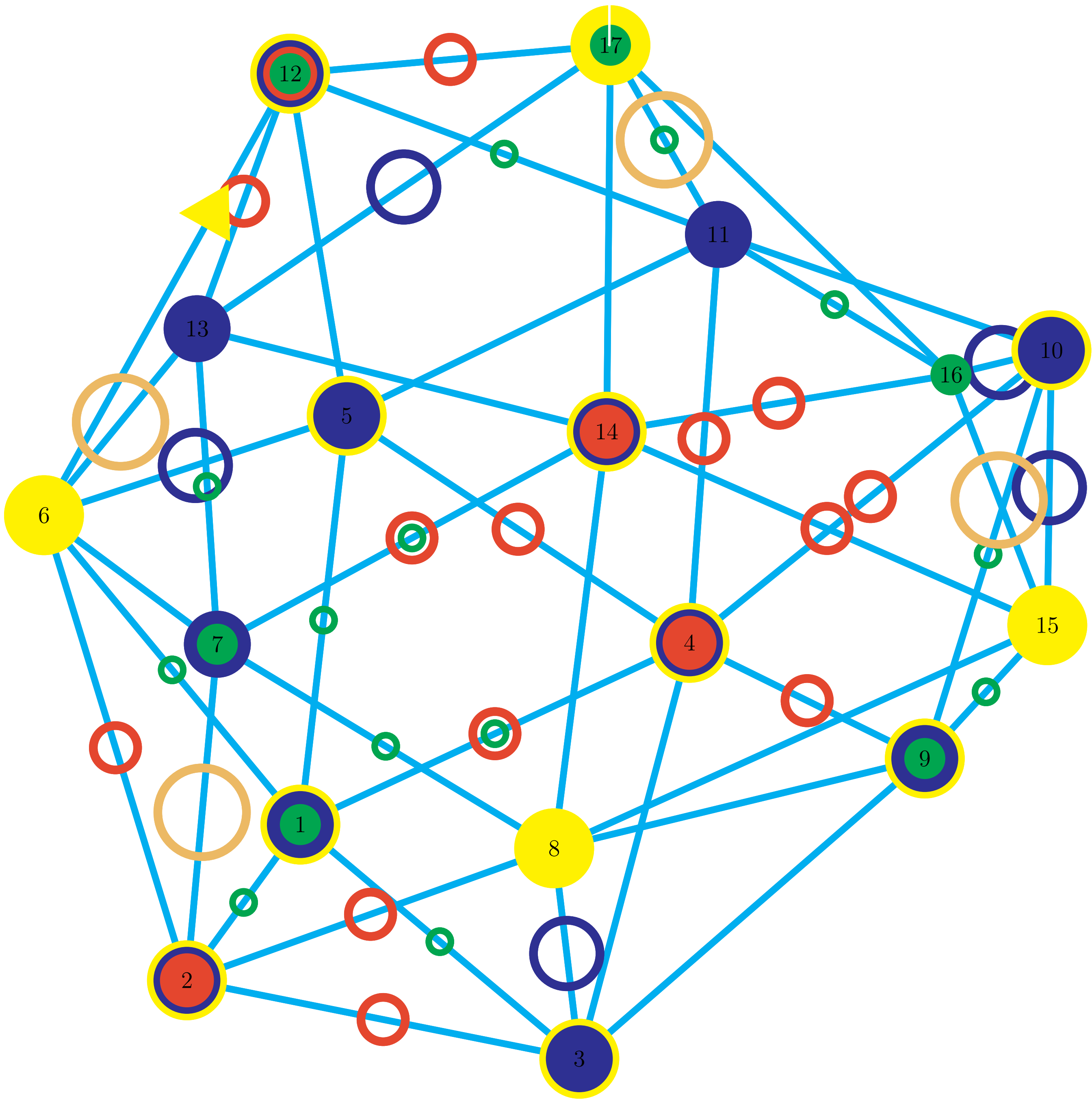

Figure 72. Instructions: 223: place edge 6→12 Yellow ArrowR,

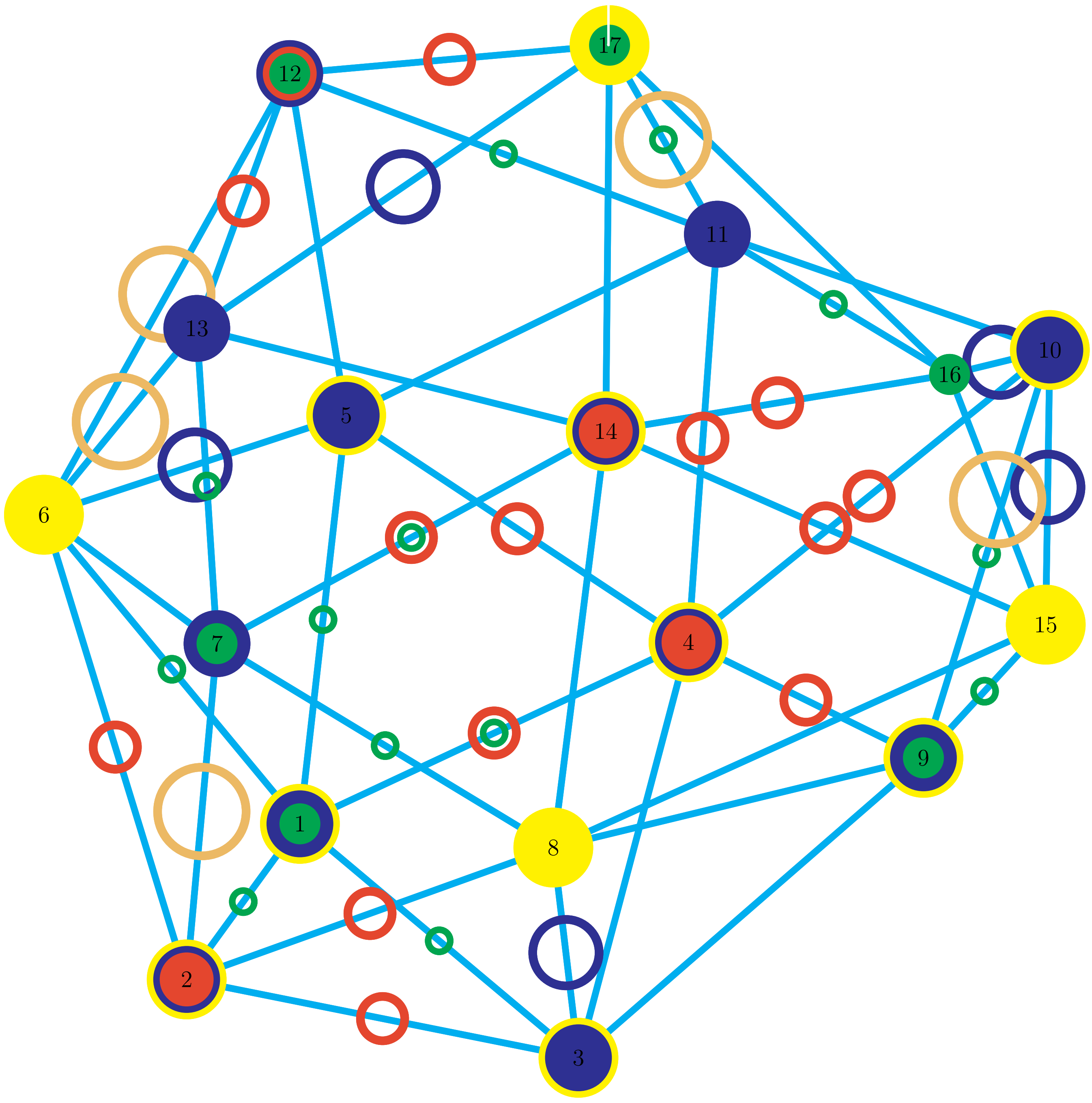

Figure 73. Instructions: 226: color edge 6–12 Yellow, 227: uncolor vertex 12 Yellow,

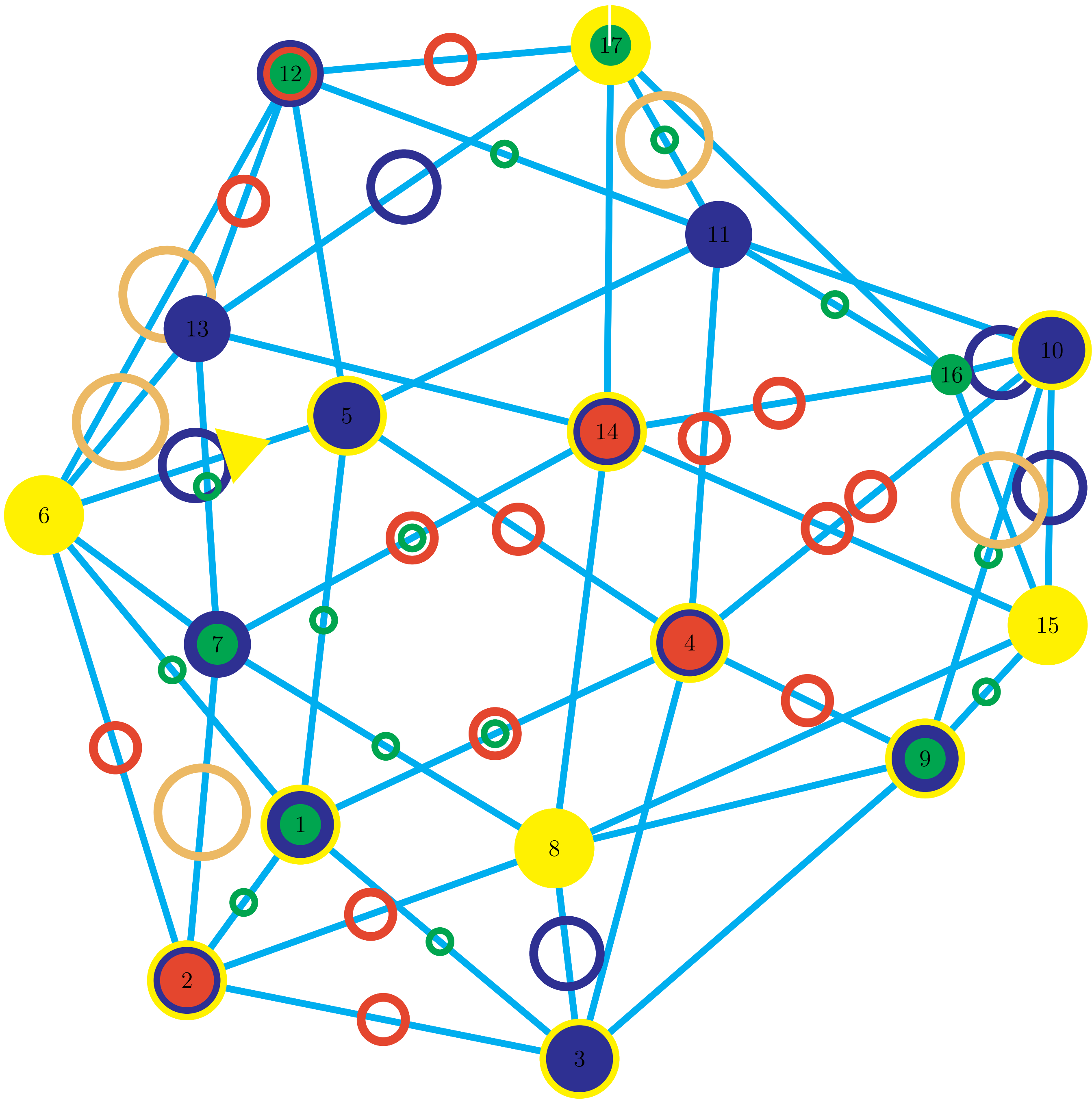

Figure 74. Instructions: 229: place edge 6→5 Yellow ArrowR,

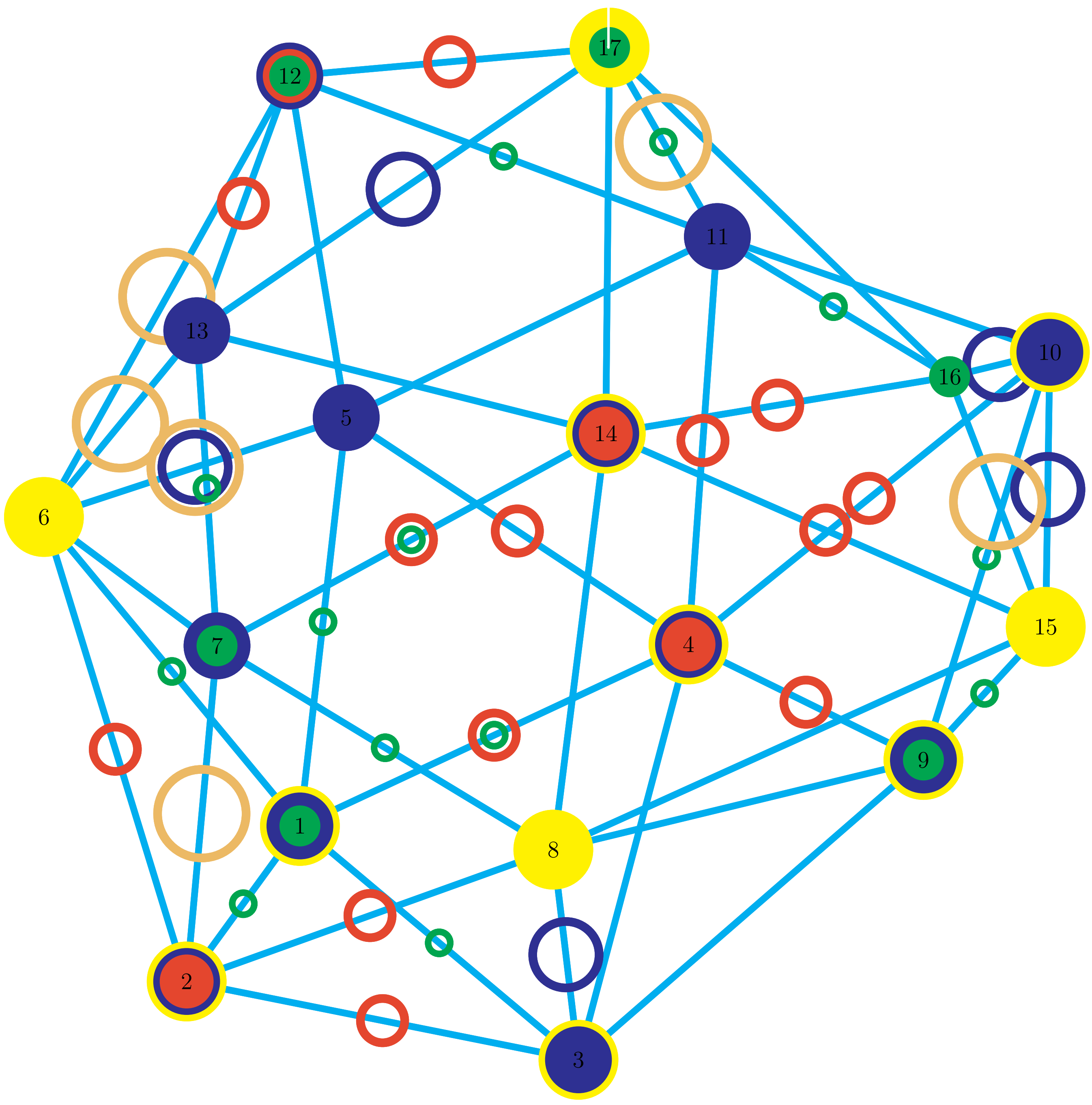

Figure 75. Instructions: 232: color edge 6–5 Yellow, 233: uncolor vertex 5 Yellow,

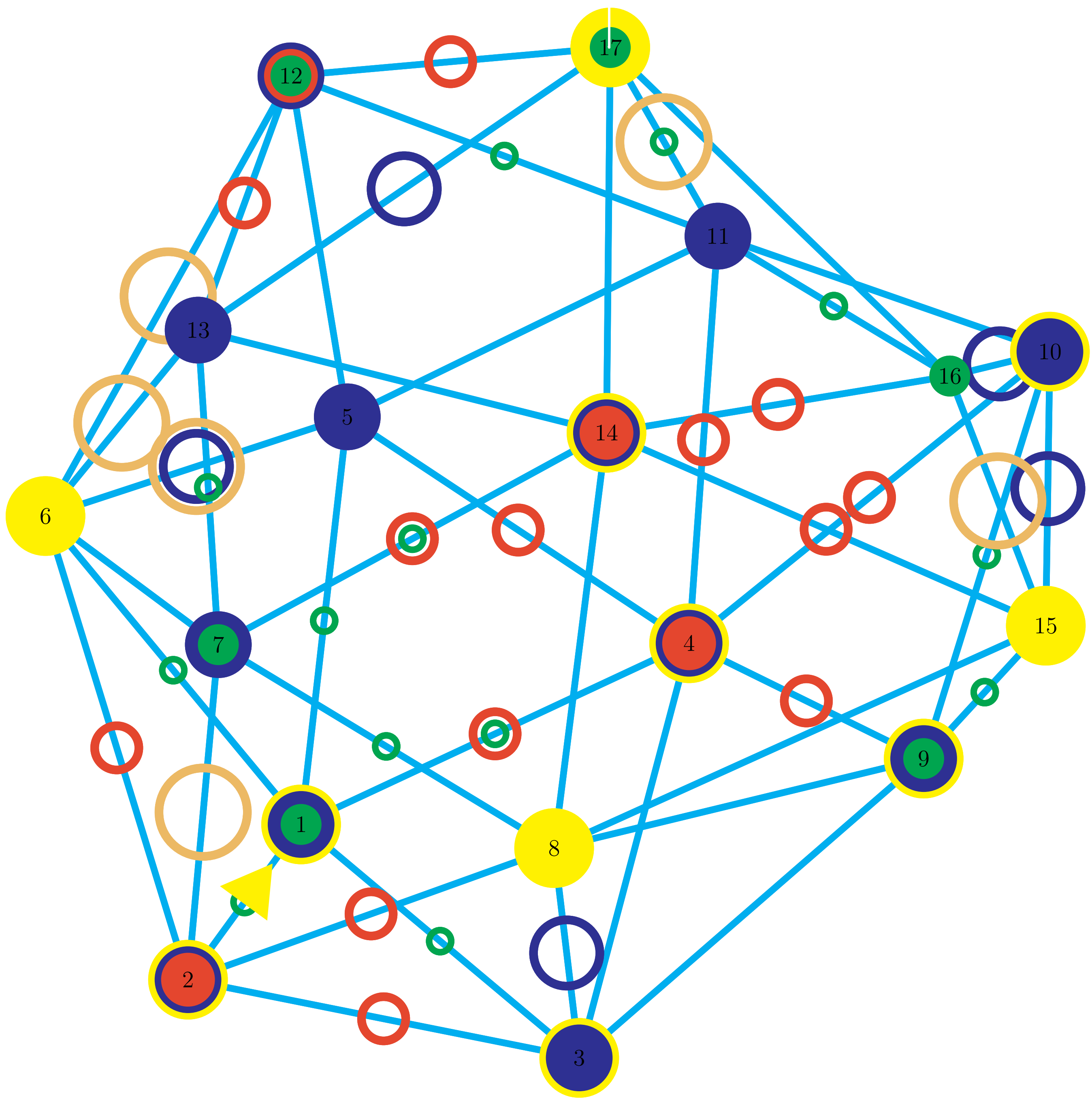

Figure 76. Instructions: 235: place edge 2→1 Yellow ArrowR,

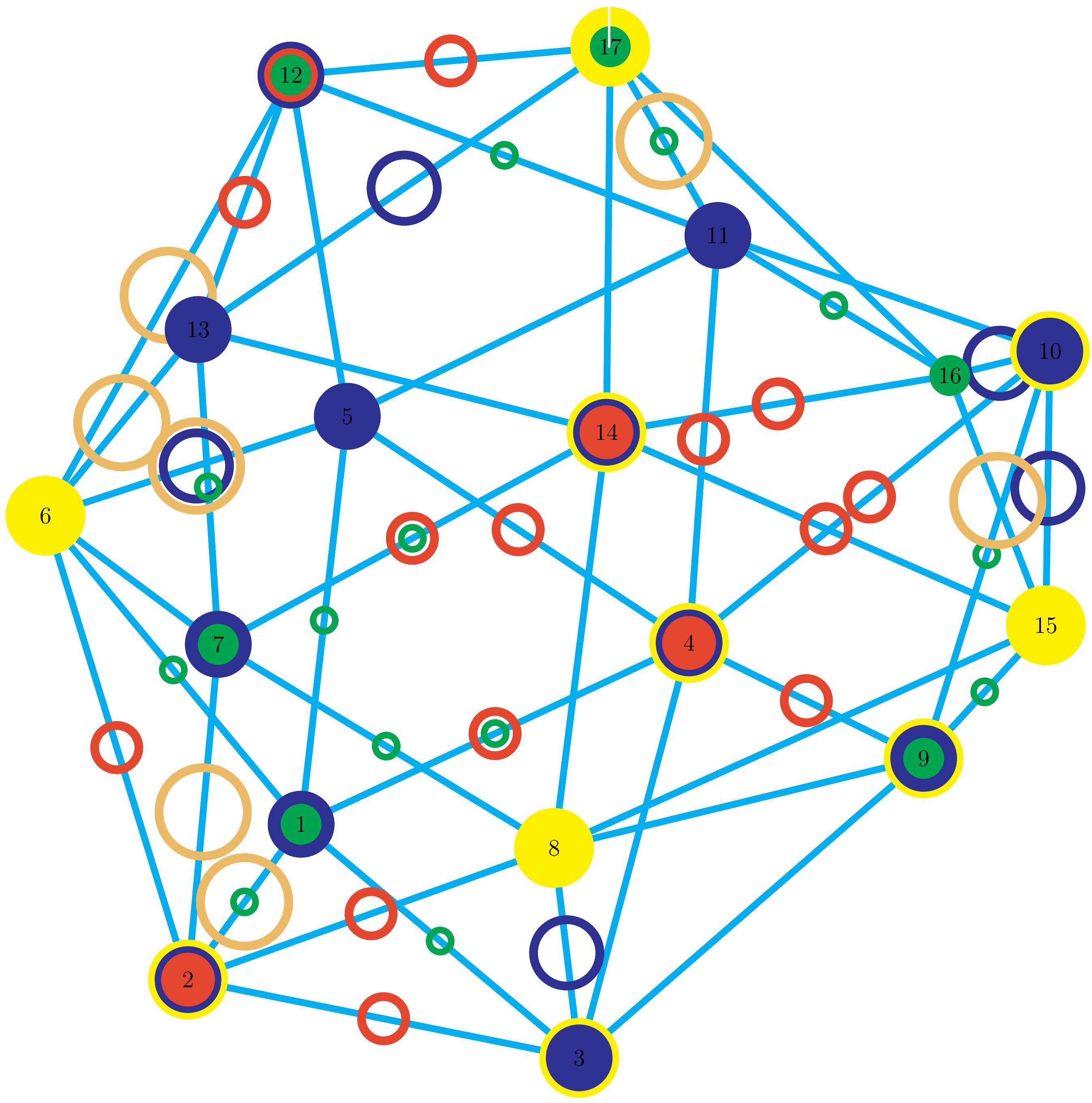

Figure 77. Instructions: 238: color edge 2–1 Yellow, 239: uncolor vertex 1 Yellow,

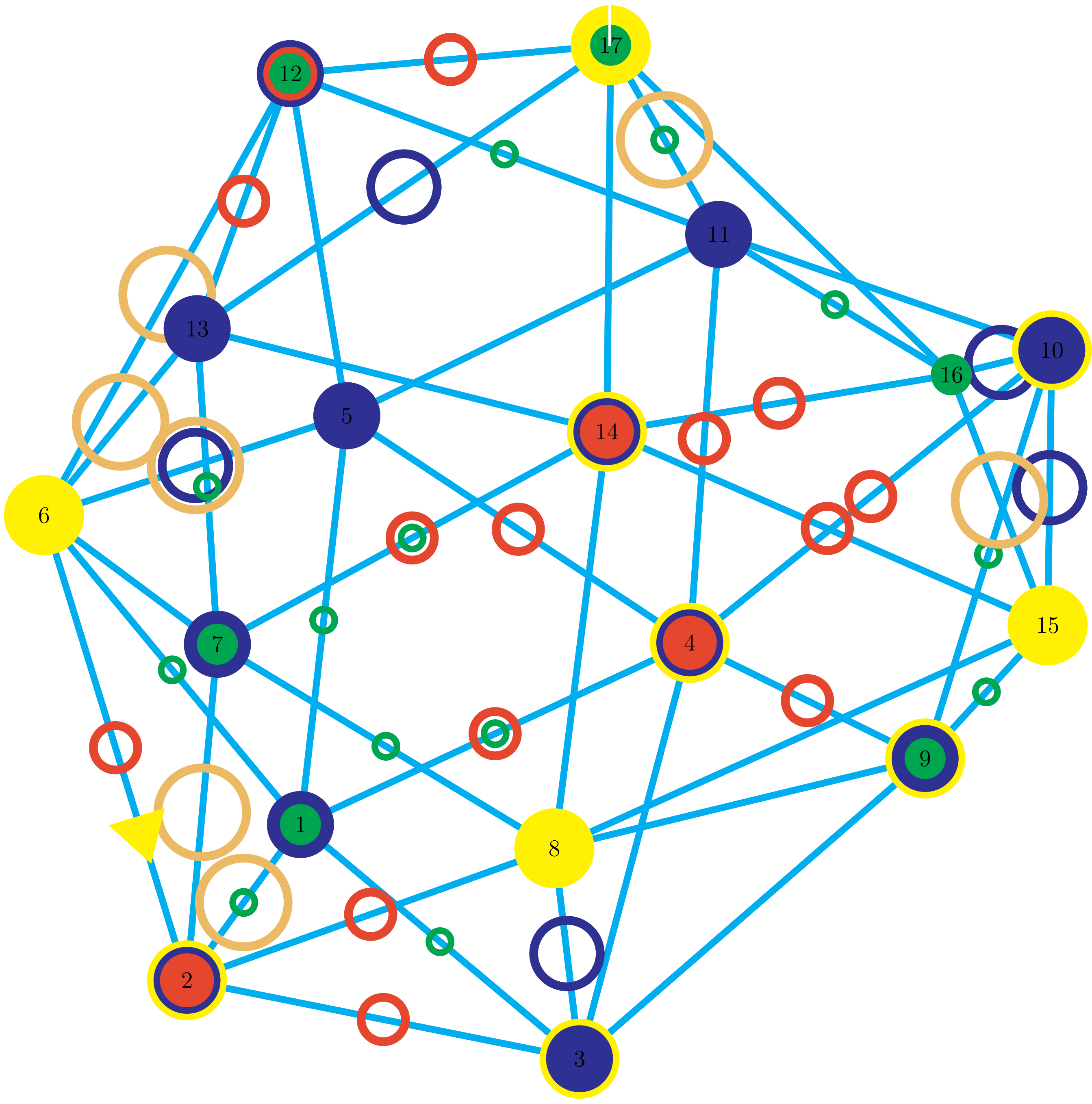

Figure 78. Instructions: 241: place edge 6→2 Yellow ArrowR,

A yellow arrow has been placed on edge 6→2 indicating removal of the yellow checker from vertex 2. This will create a new problem.

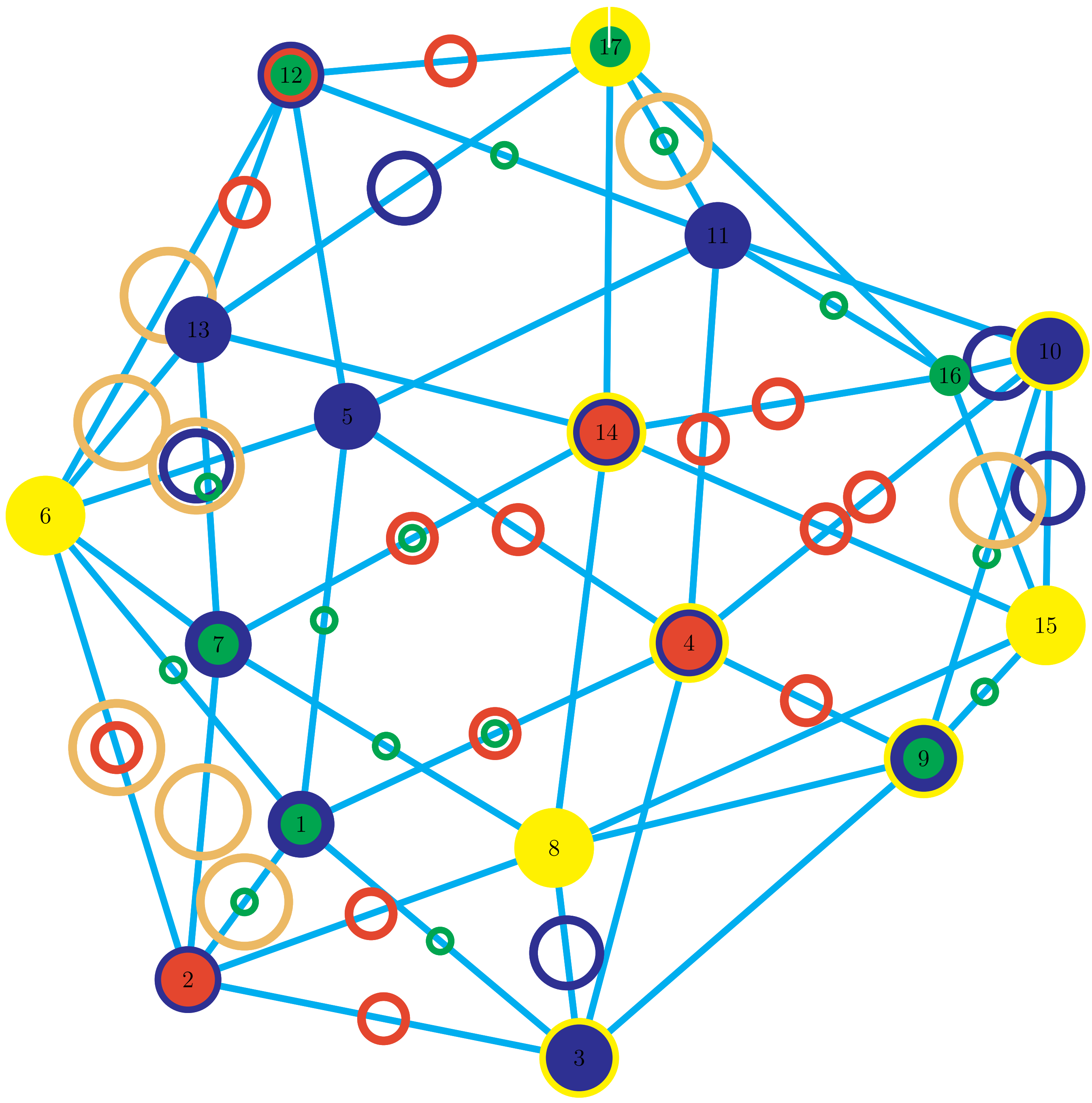

Figure 79. Instructions: 244: color edge 6–2 Yellow, 245: uncolor vertex 2 Yellow,

The yellow checker has been removed from vertex 2 and a yellow circle has been placed on edge 2-6. Now property A obtains but property B does not for there are yellow circles on edges 1-2 and 2-7 but no yellow checkers on vertices 1, 2 or 7. However the situation is easily remedied as follows.

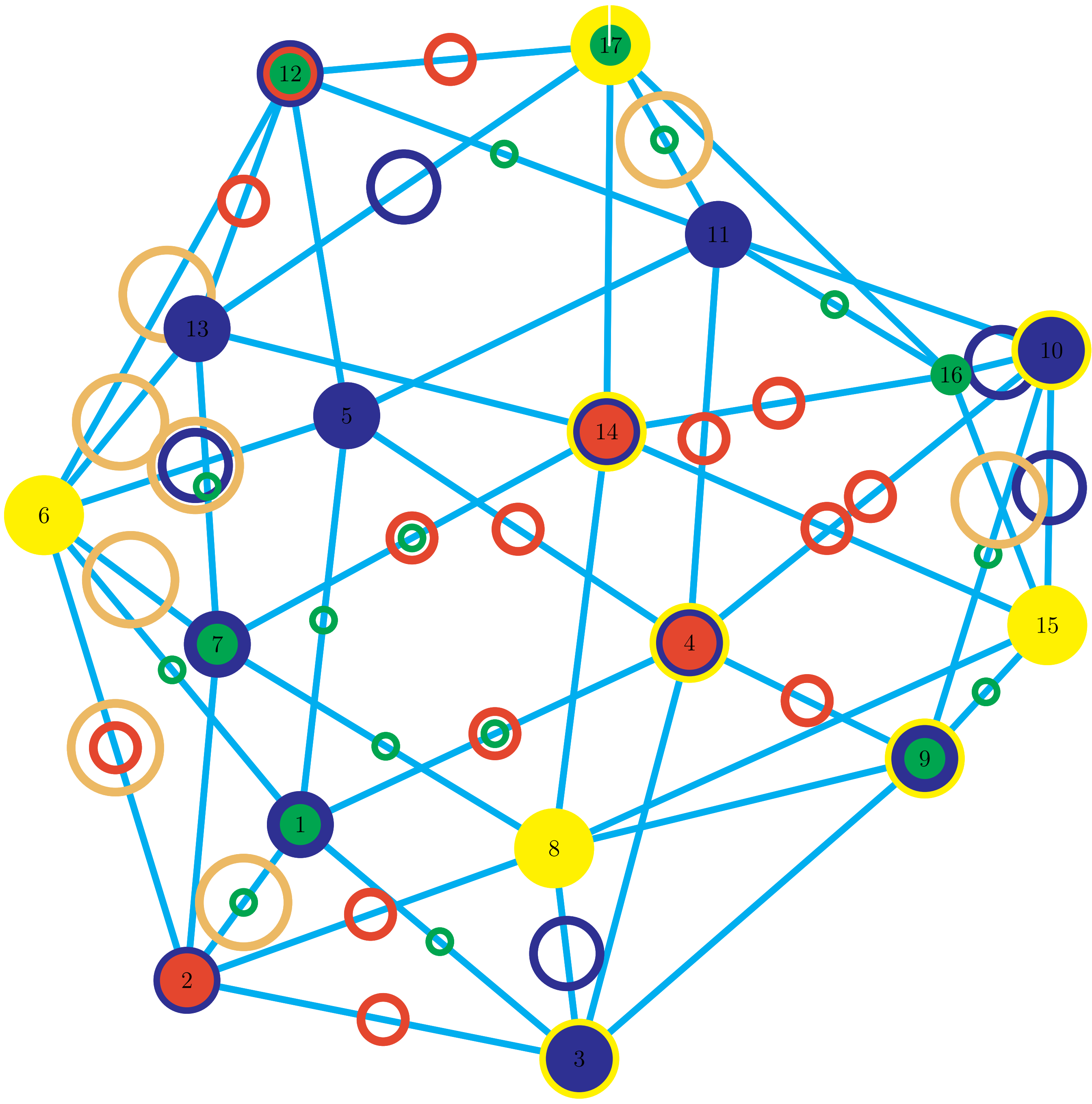

Figure 80. Instructions: 249: color edge 6–7 Yellow,

The yellow circle on edge 2-7 has been flipped to edge 6-7.

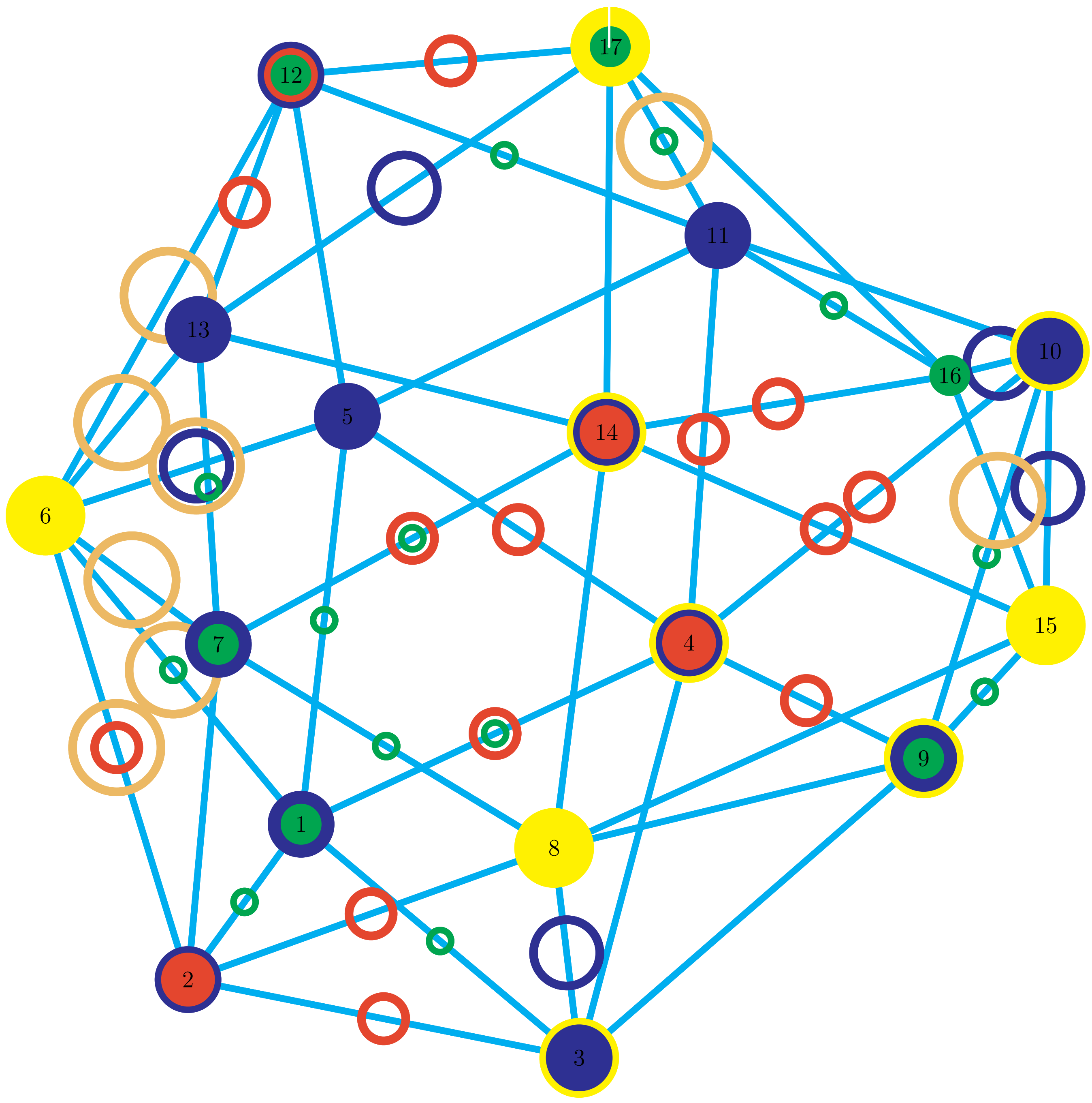

Figure 81. Instructions: 253: color edge 6–1 Yellow,

The yellow circle on edge 1-2 has been flipped to edge 2-6 and property B is restored.

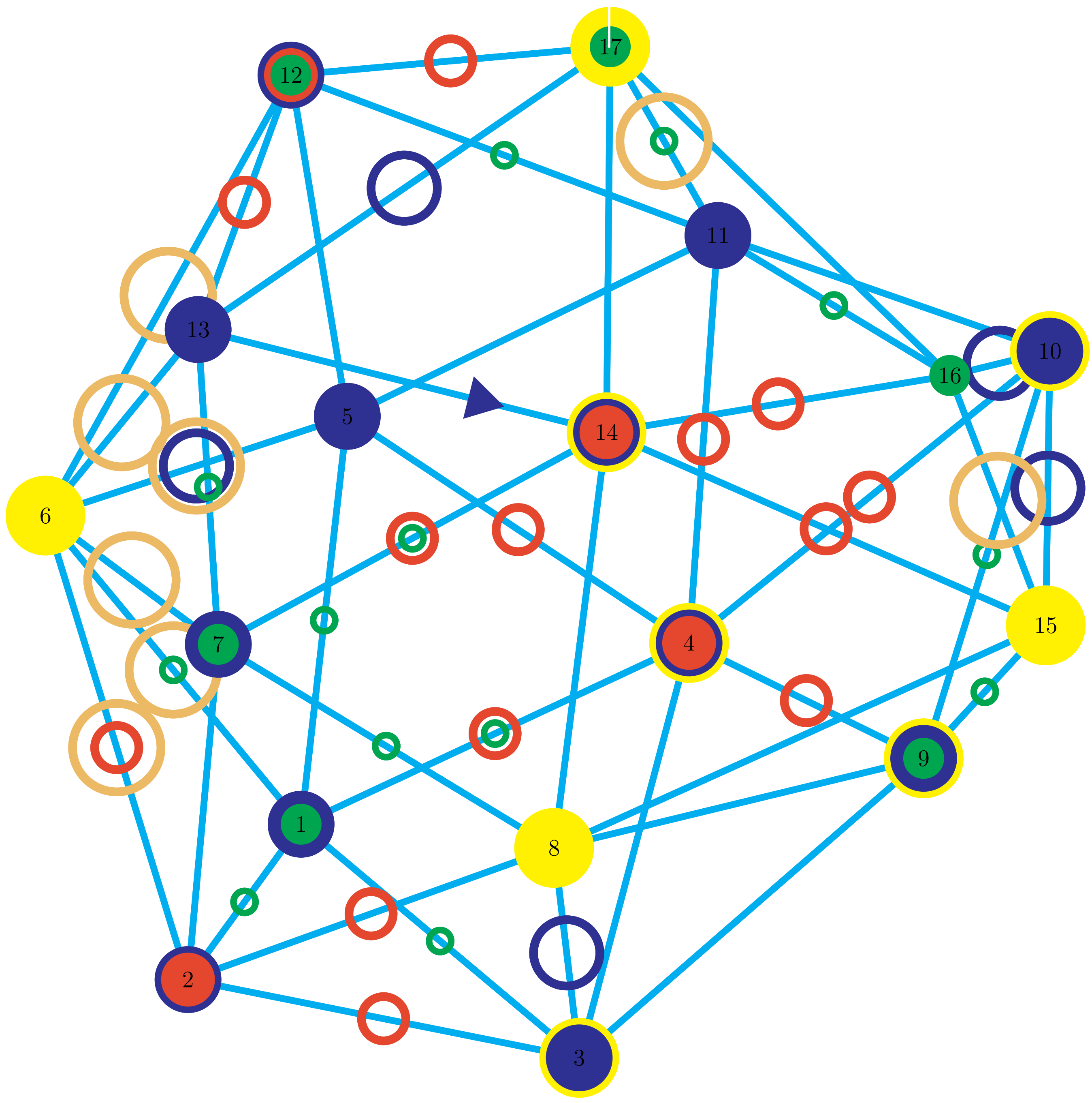

Figure 82. Instructions: 255: place edge 13→14 Blue ArrowR,

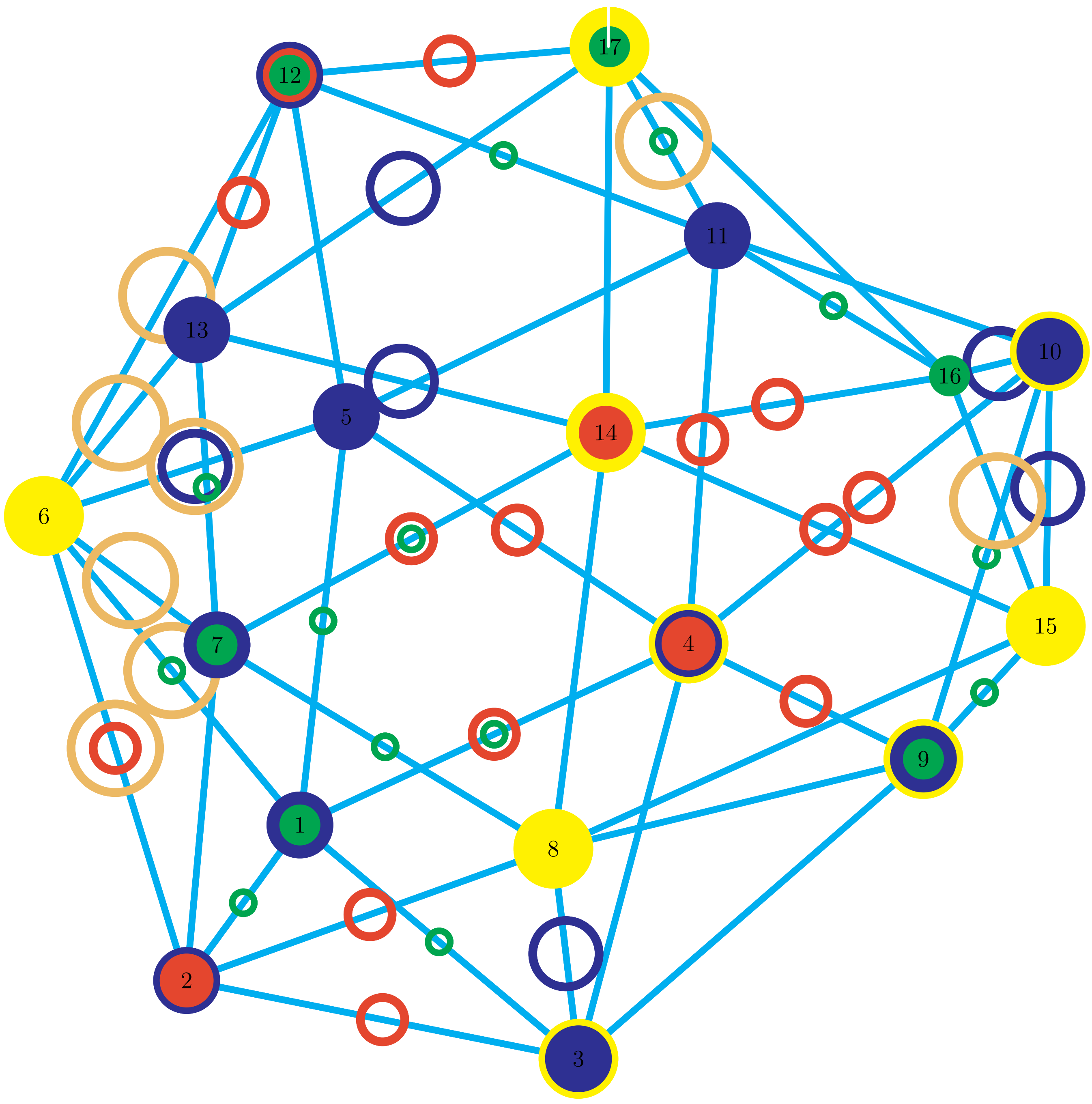

Figure 83. Instructions: 258: color edge 13–14 Blue, 259: uncolor vertex 14 Blue,

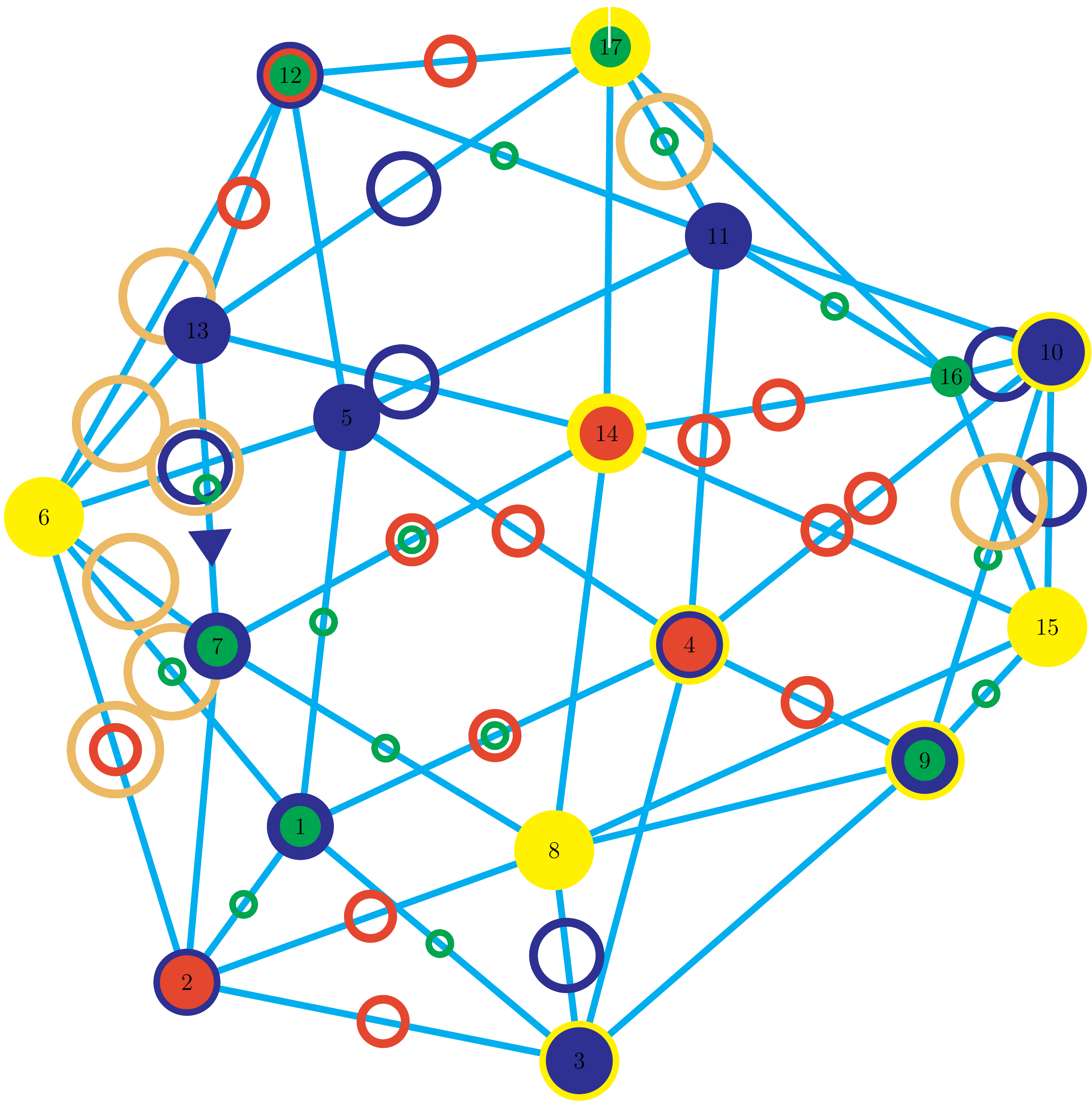

Figure 84. Instructions: 261: place edge 13→7 Blue ArrowR,

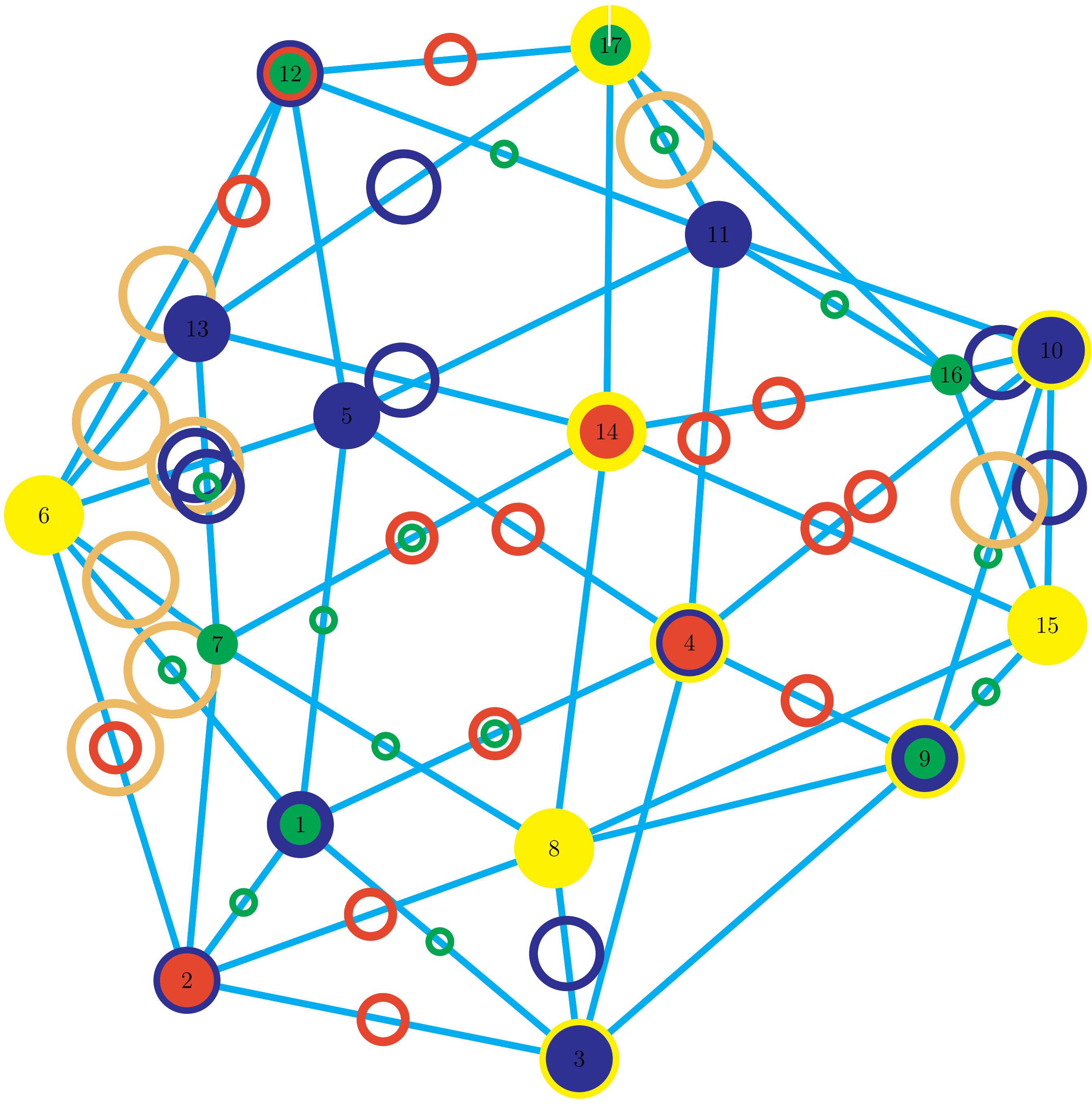

Figure 85. Instructions: 264: color edge 13–7 Blue, 265: uncolor vertex 7 Blue,

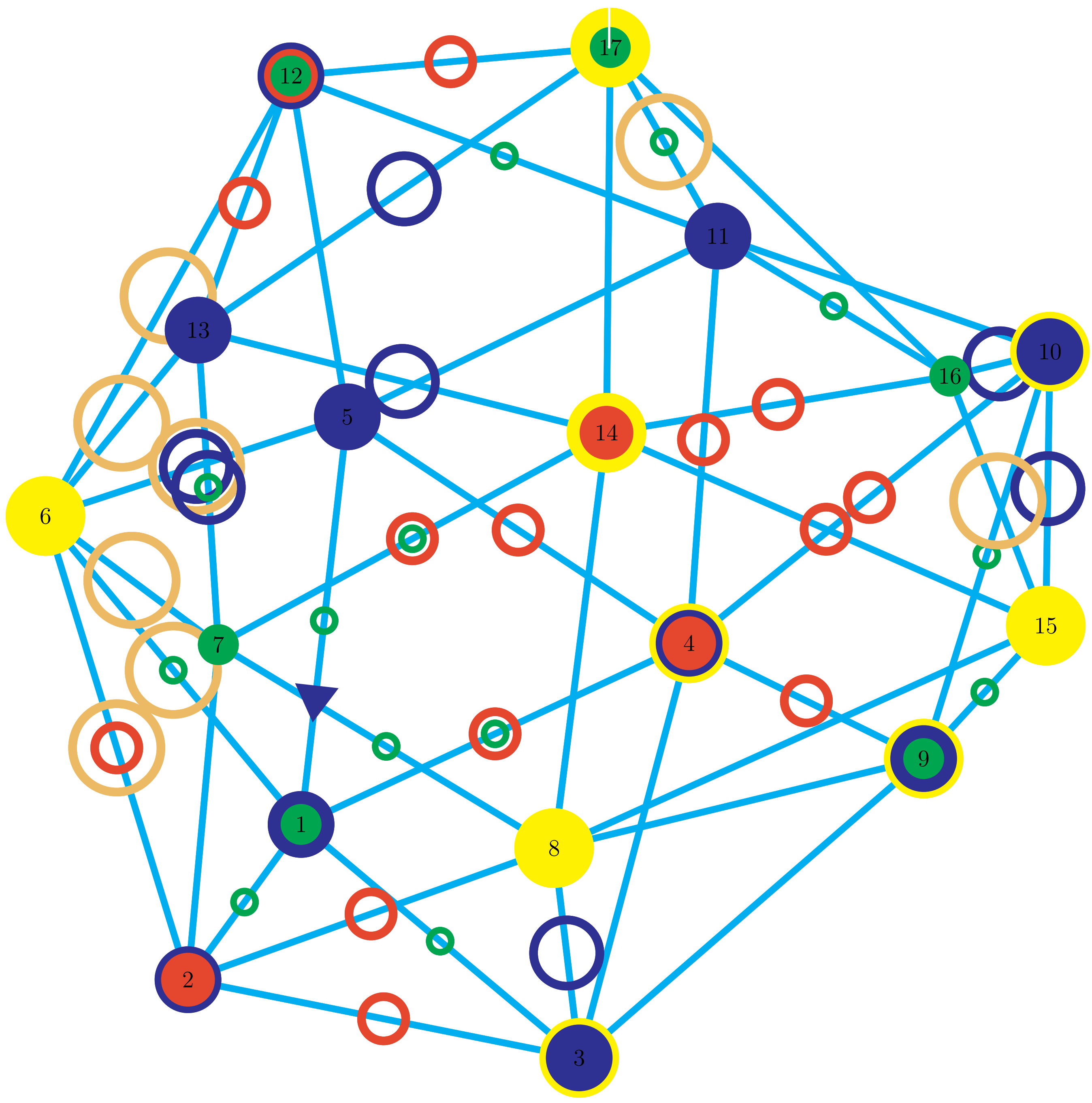

Figure 86. Instructions: 267: place edge 5→1 Blue ArrowR,

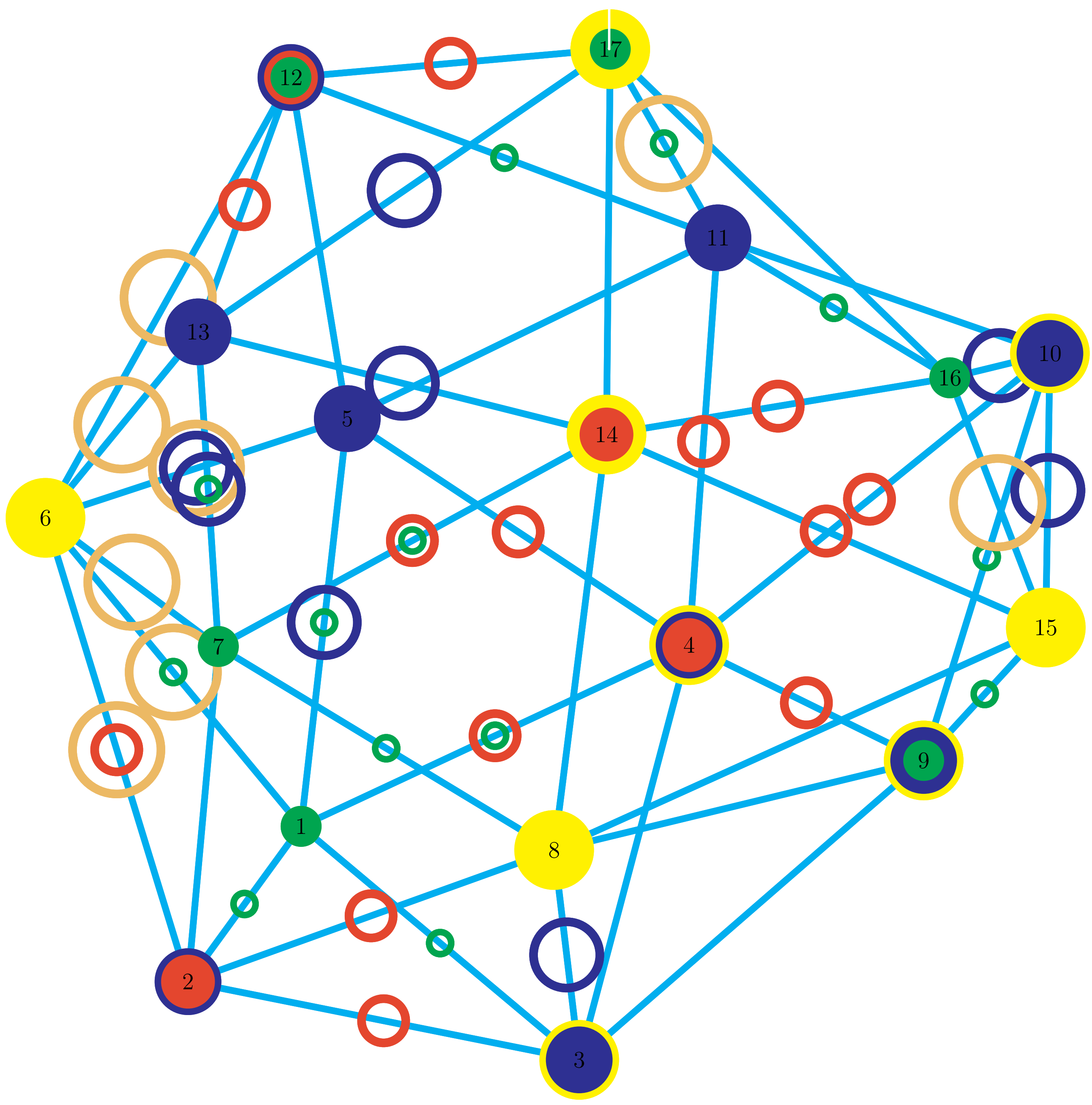

Figure 87. Instructions: 270: color edge 5–1 Blue, 271: uncolor vertex 1 Blue,

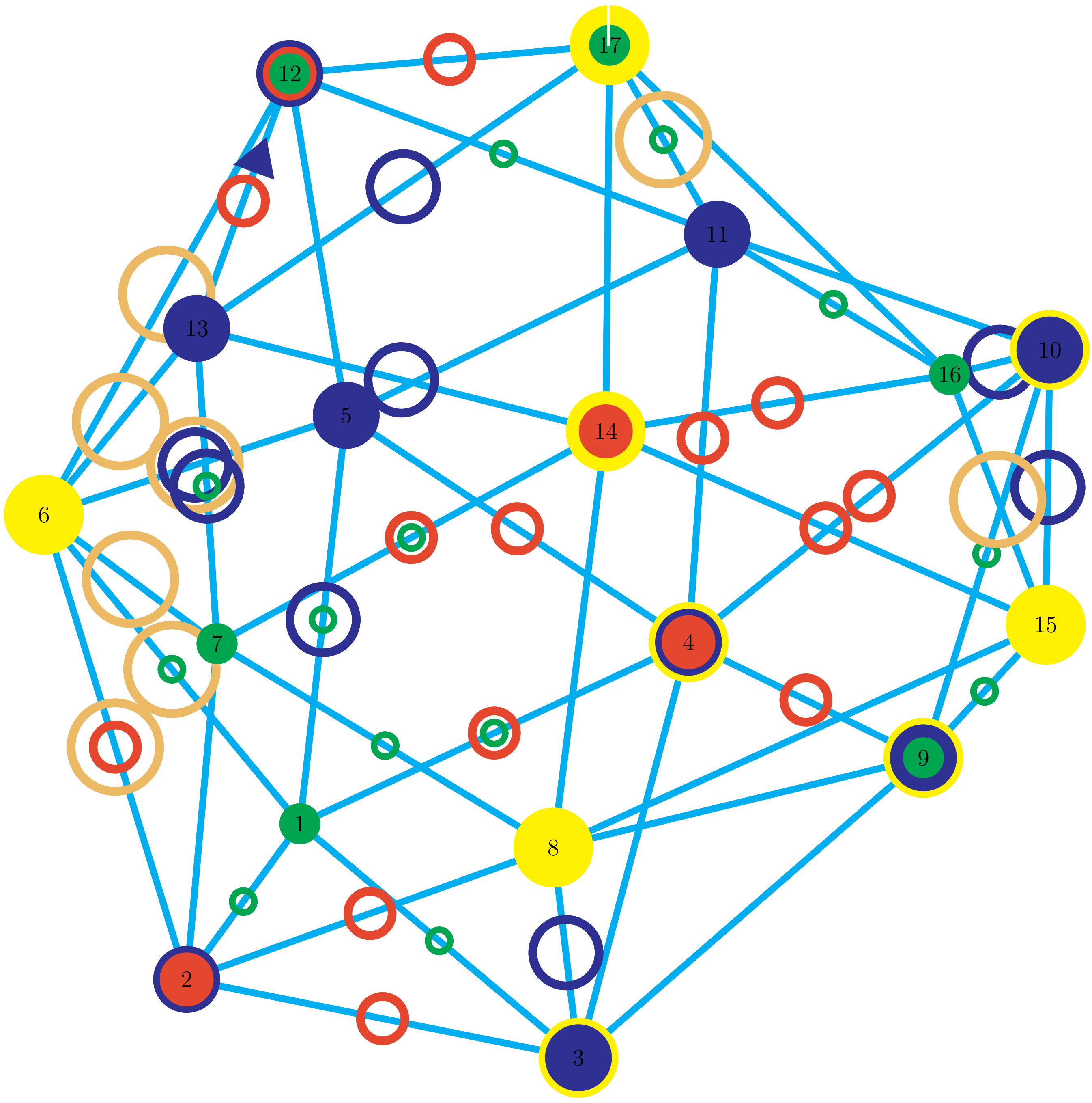

Figure 88. Instructions: 273: place edge 13→12 Blue ArrowR,

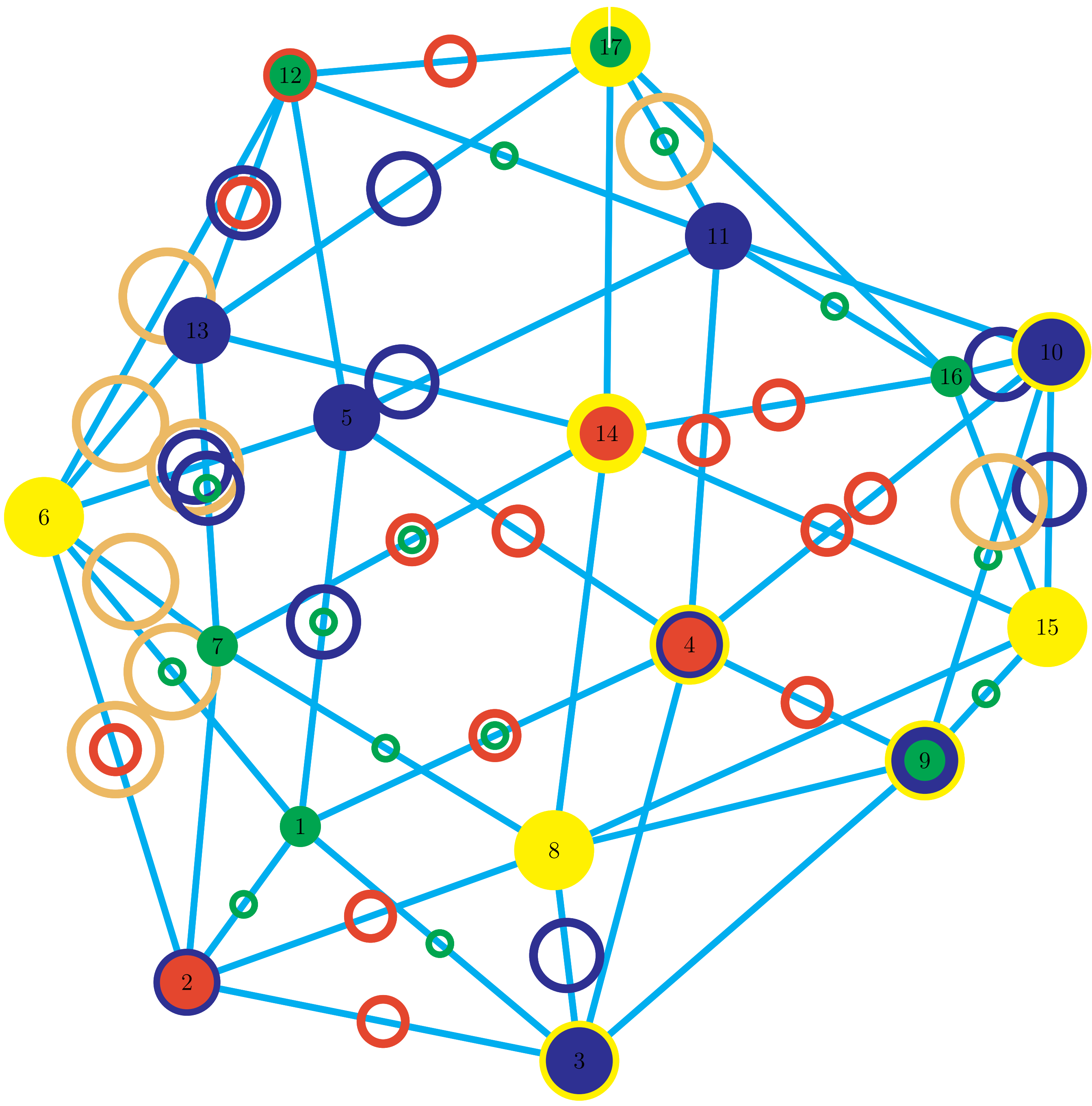

Figure 89. Instructions: 276: color edge 13–12 Blue, 277: uncolor vertex 12 Blue,

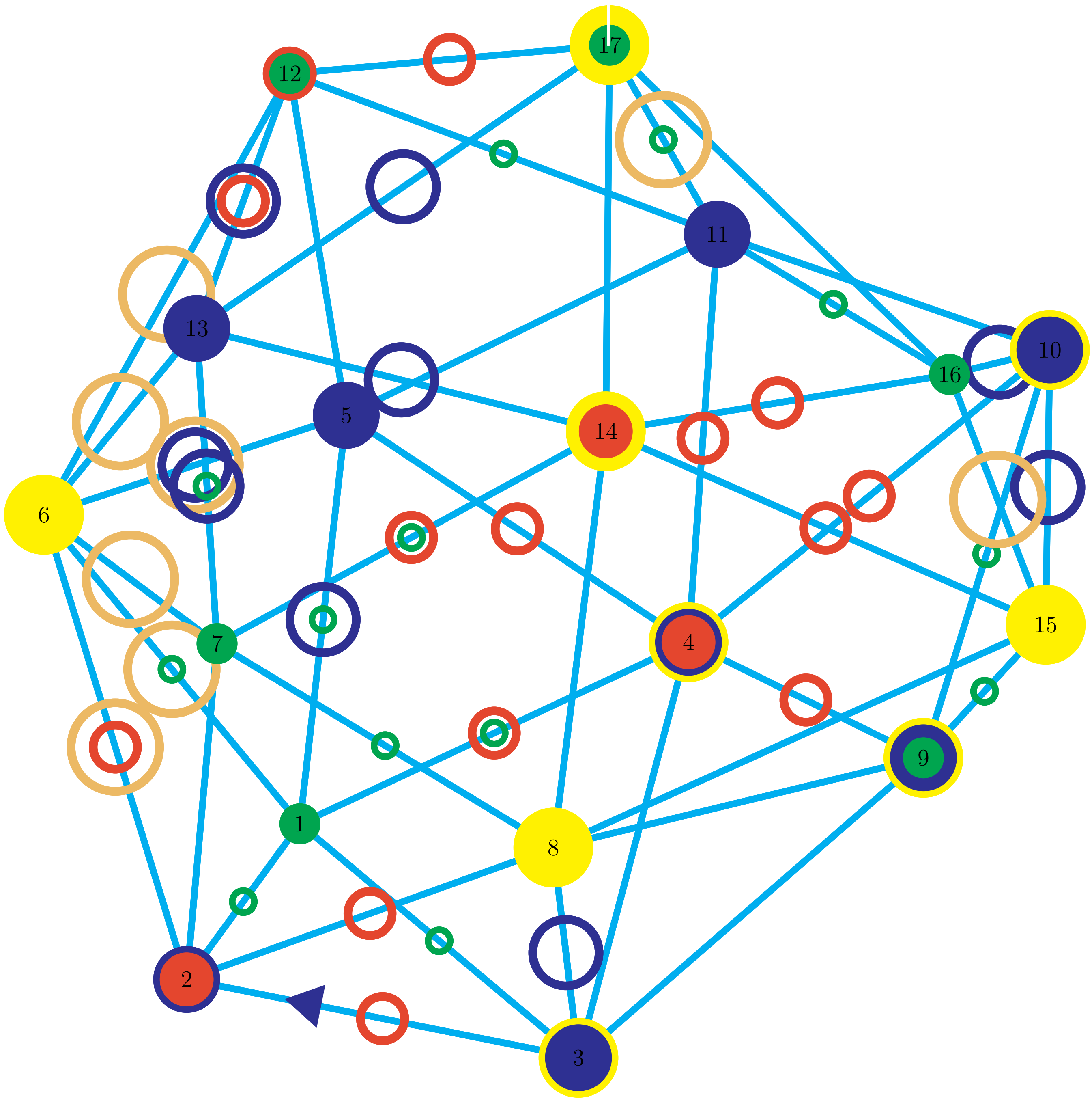

Figure 90. Instructions: 279: place edge 3→2 Blue ArrowR,

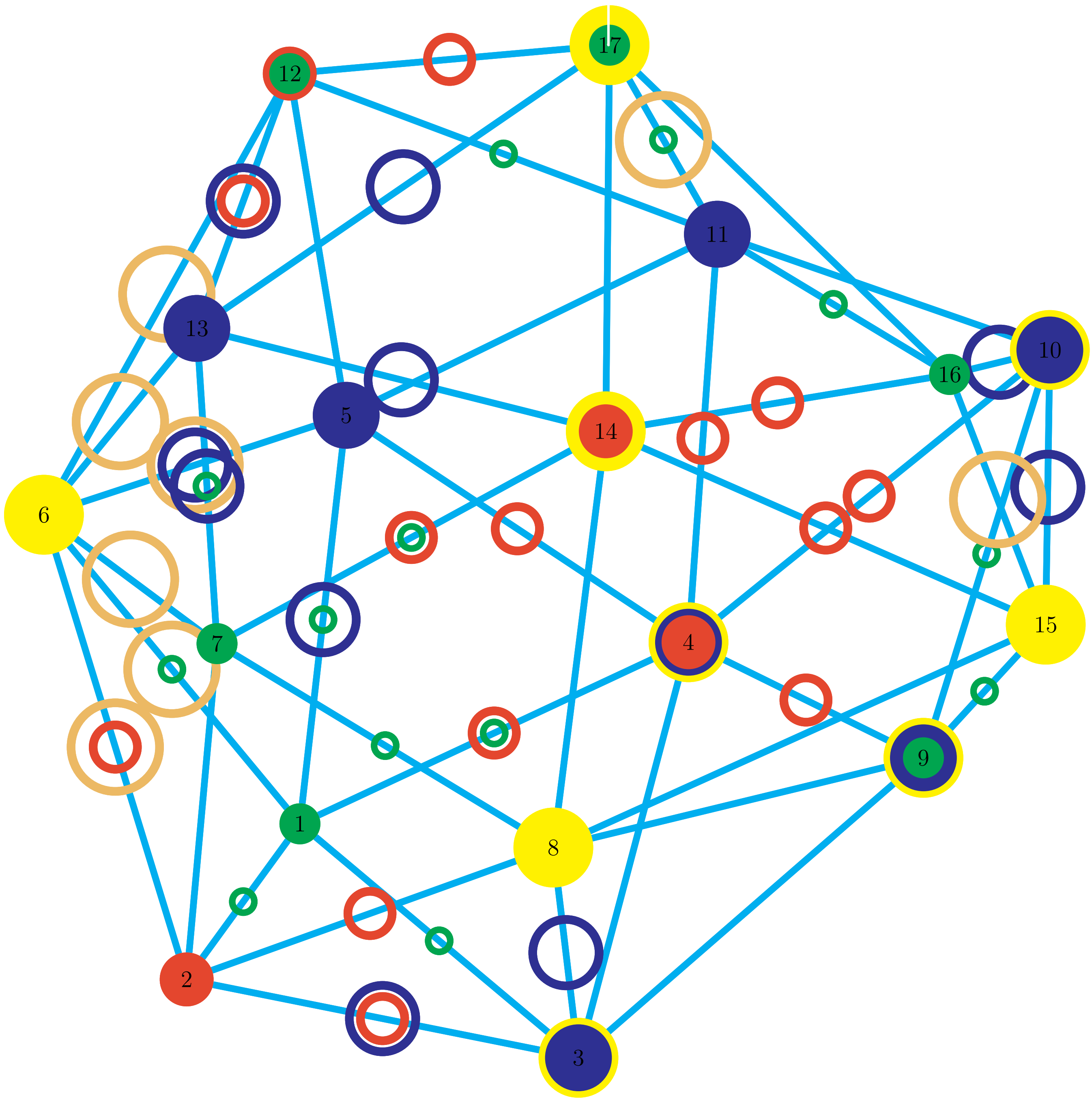

Figure 91. Instructions: 282: color edge 3–2 Blue, 283: uncolor vertex 2 Blue,

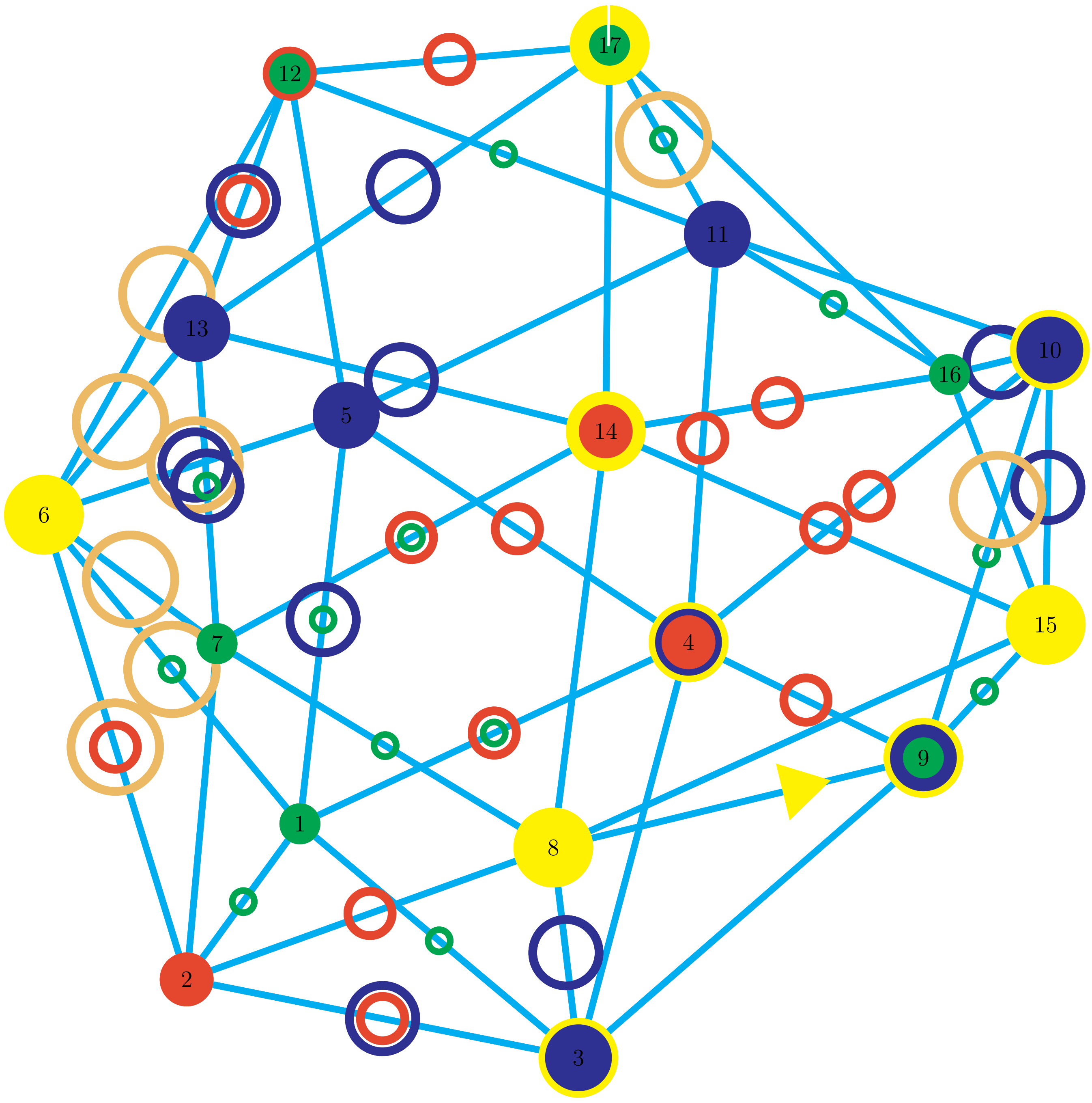

Figure 92. Instructions: 285: place edge 8→9 Yellow ArrowR,

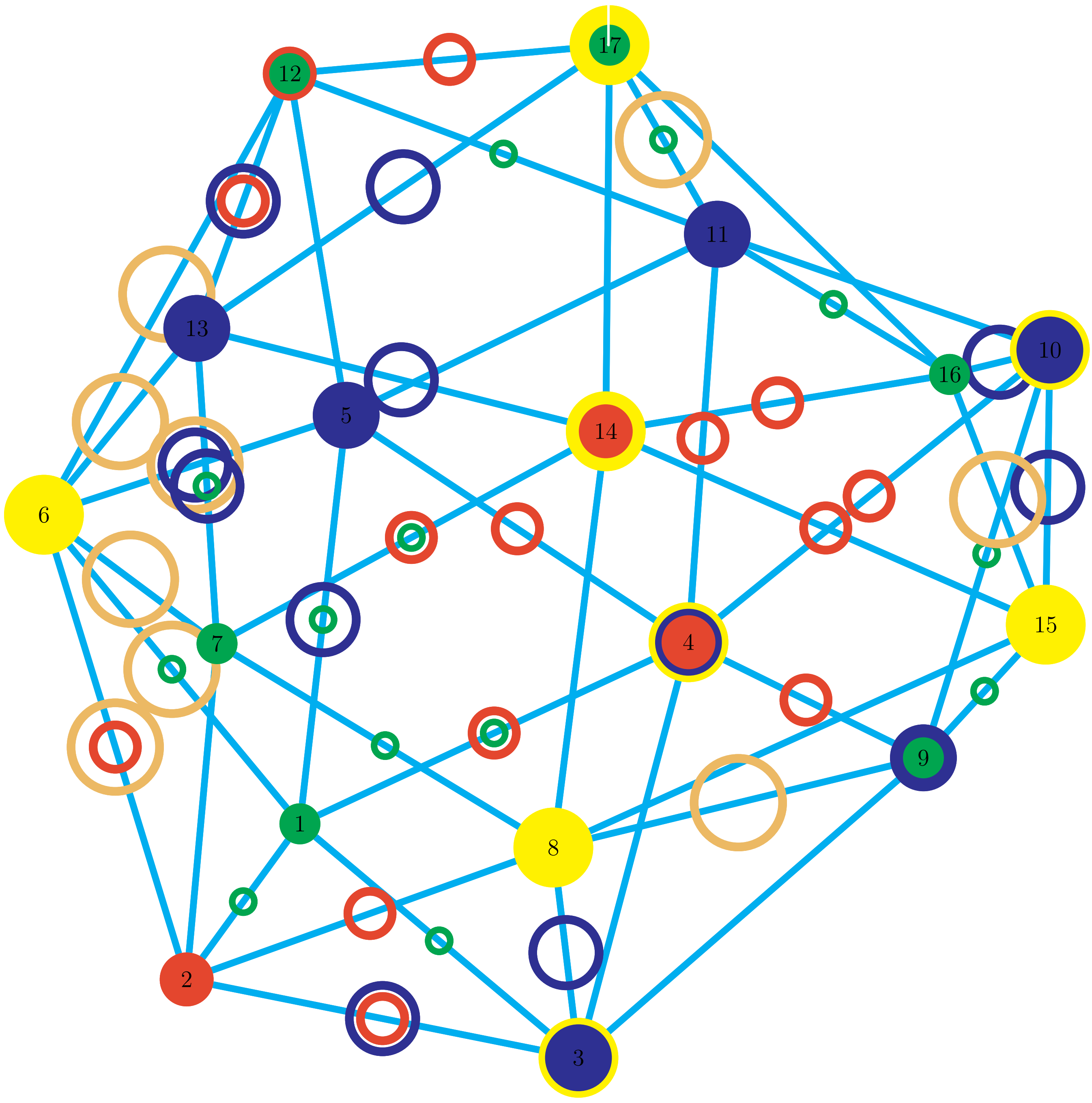

Figure 93. Instructions: 288: color edge 8–9 Yellow, 289: uncolor vertex 9 Yellow,

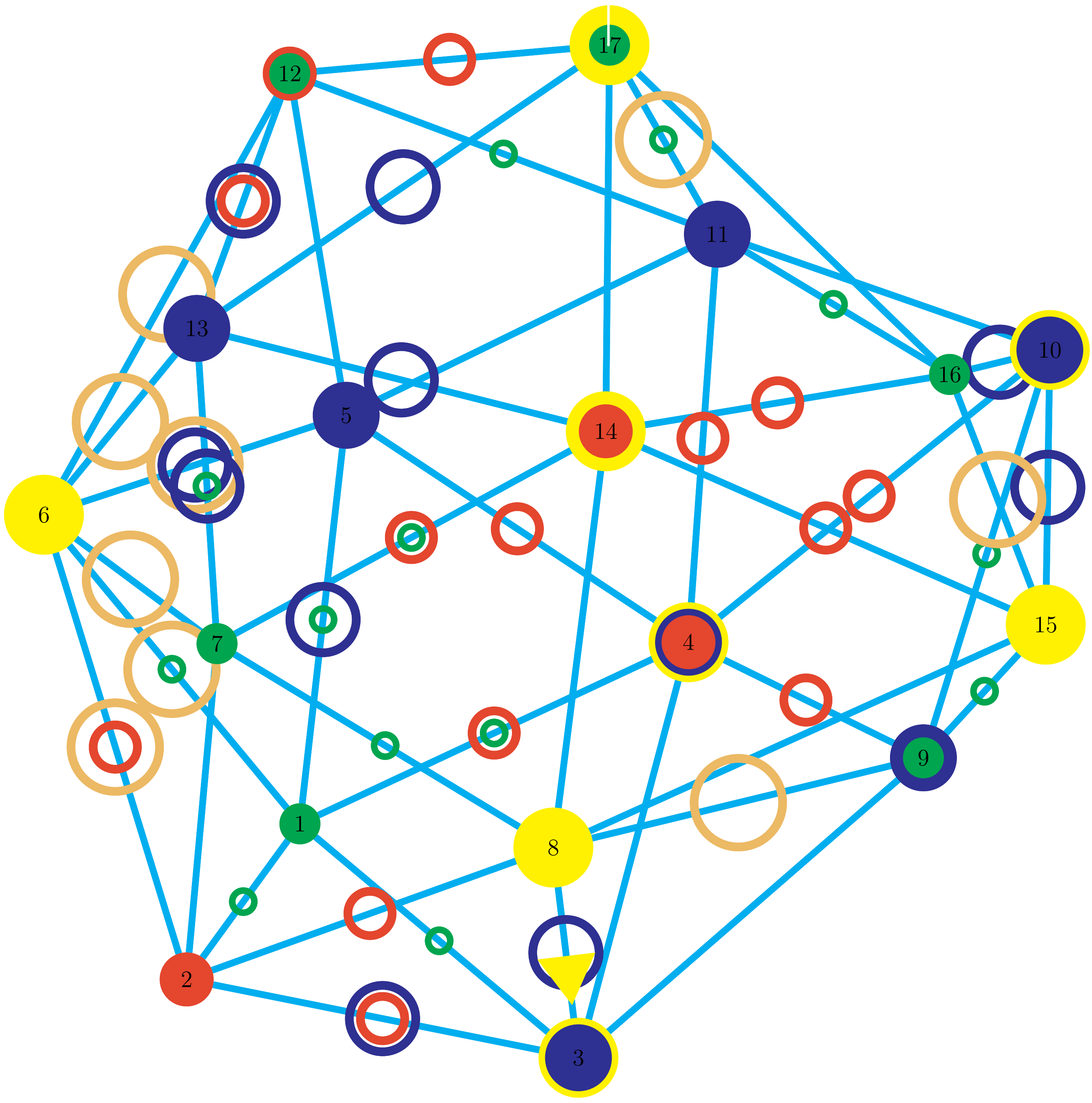

Figure 94. Instructions: 291: place edge 8→3 Yellow ArrowR,

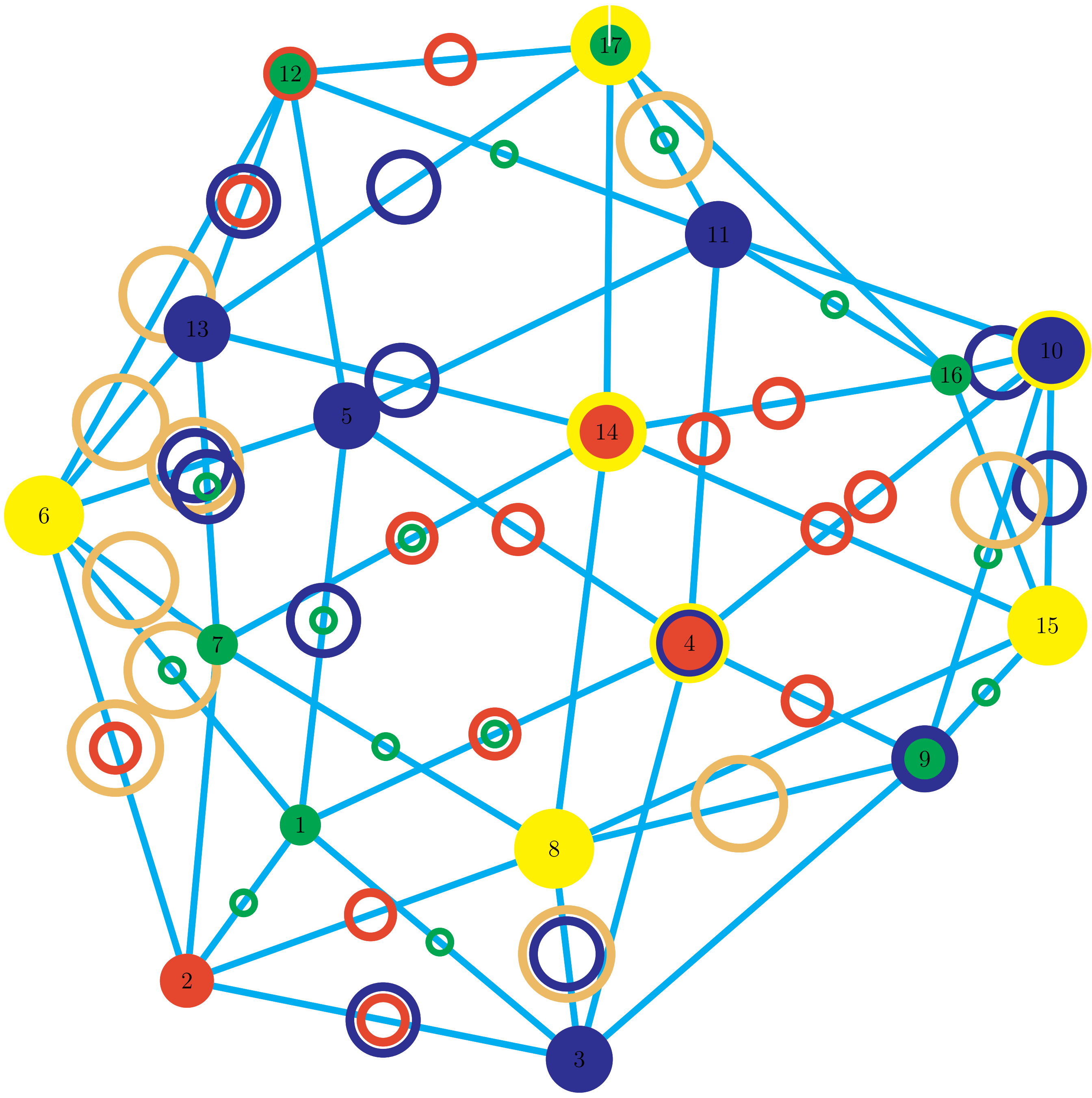

Figure 95. Instructions: 294: color edge 8–3 Yellow, 295: uncolor vertex 3 Yellow,

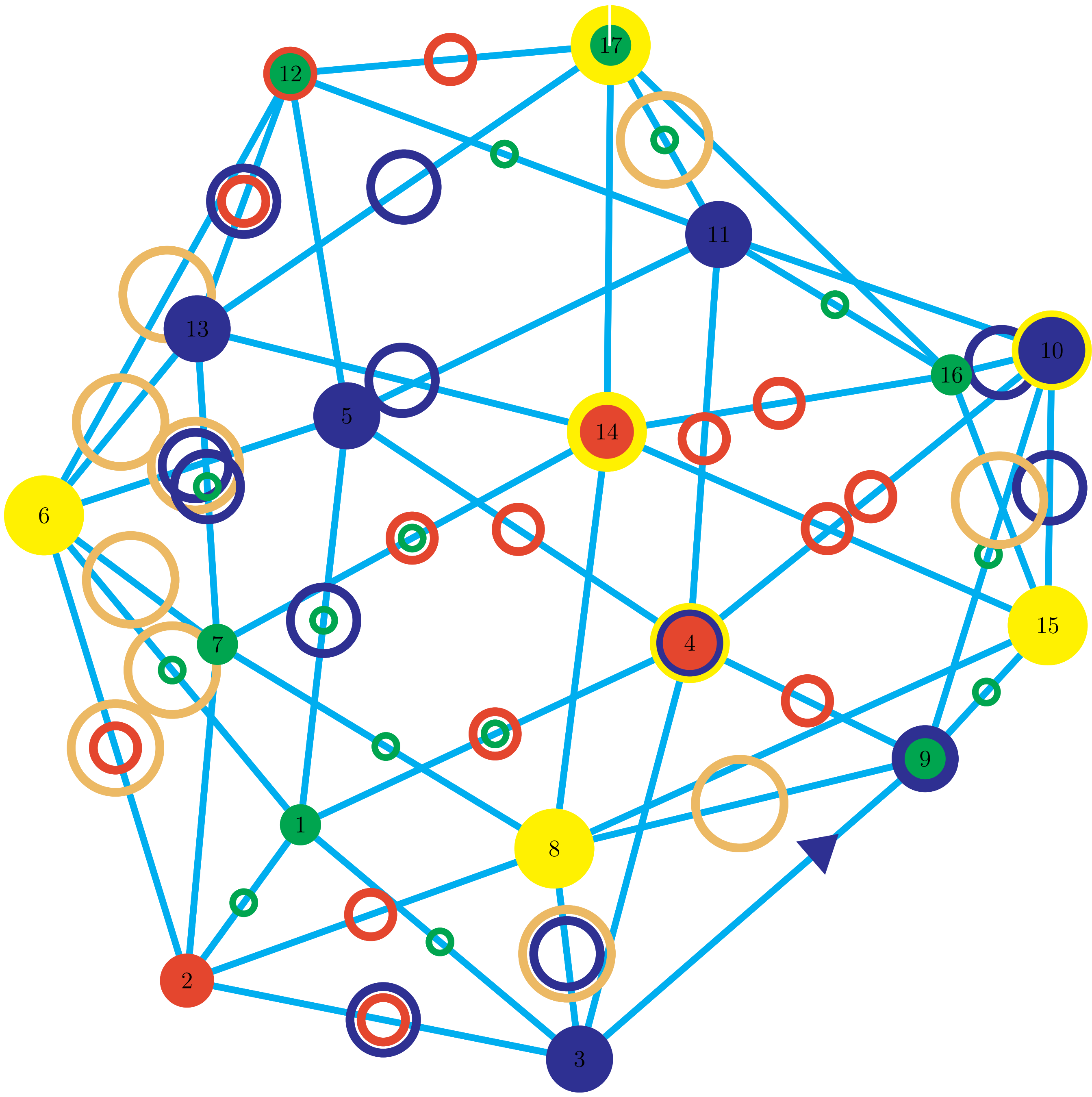

Figure 96. Instructions: 297: place edge 3→9 Blue ArrowR,

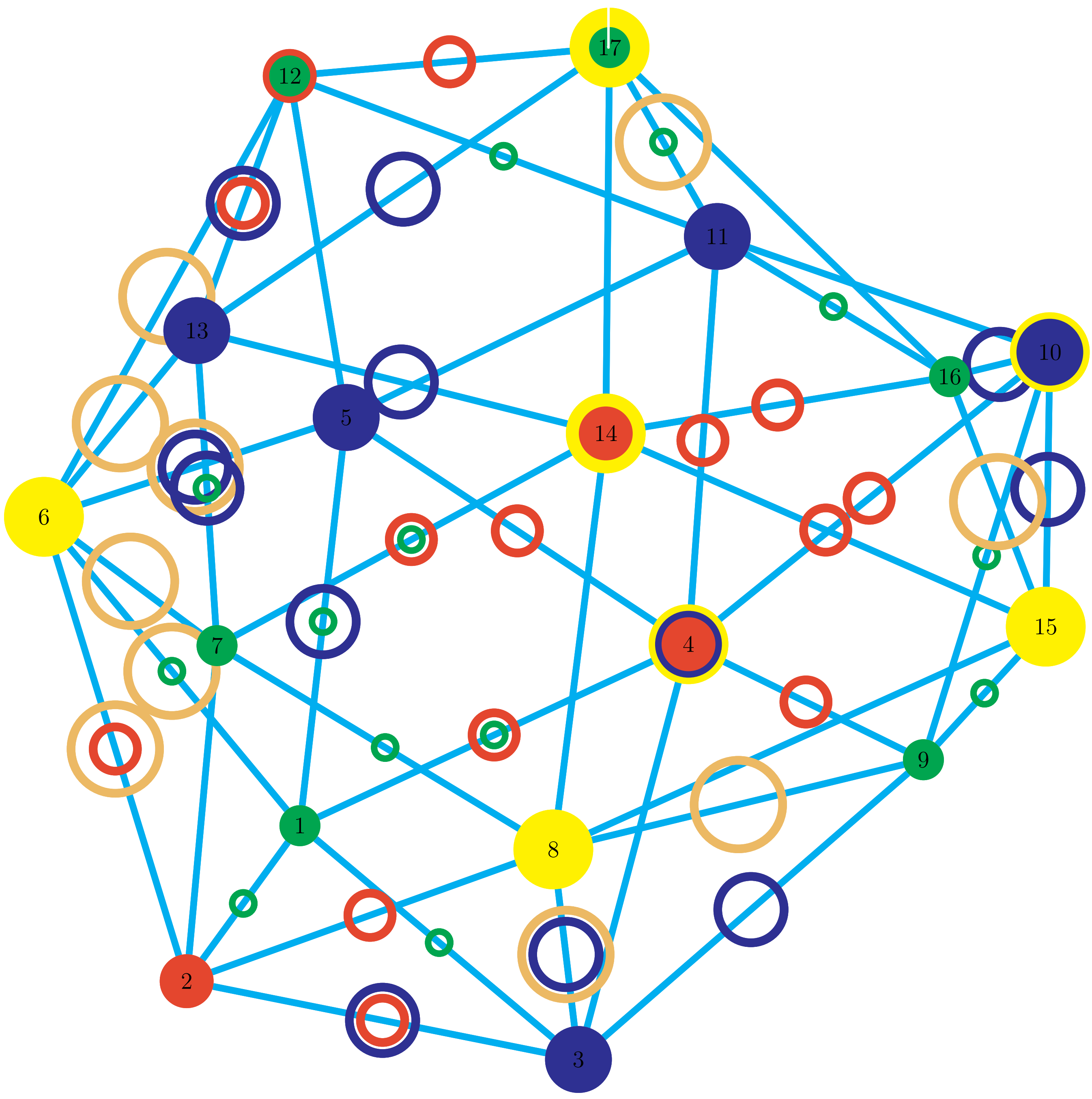

Figure 97. Instructions: 300: color edge 3–9 Blue, 301: uncolor vertex 9 Blue,

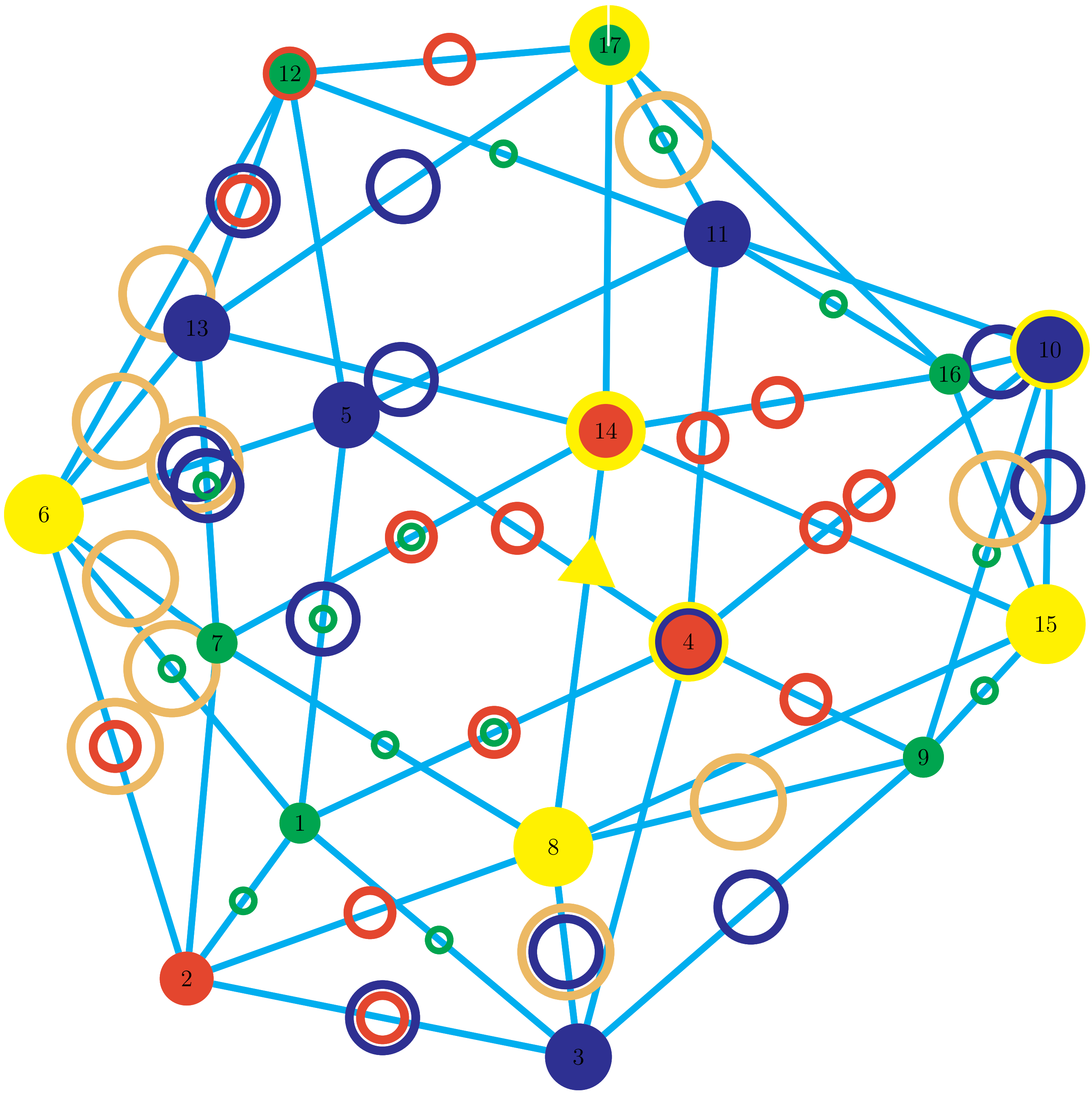

Figure 98. Instructions: 303: place edge 8→14 Yellow ArrowR,

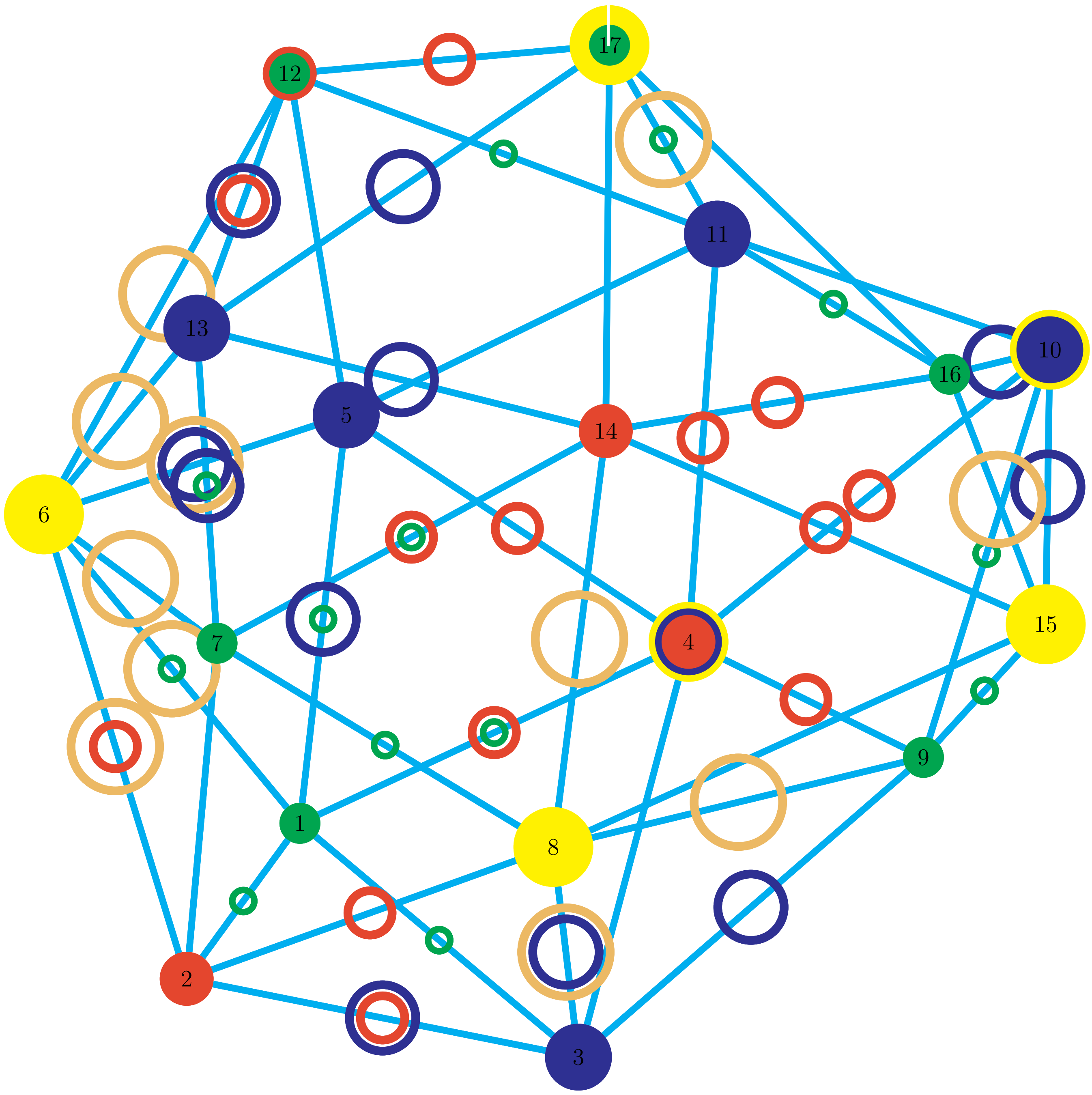

Figure 99. Instructions: 306: color edge 8–14 Yellow, 307: uncolor vertex 14 Yellow,

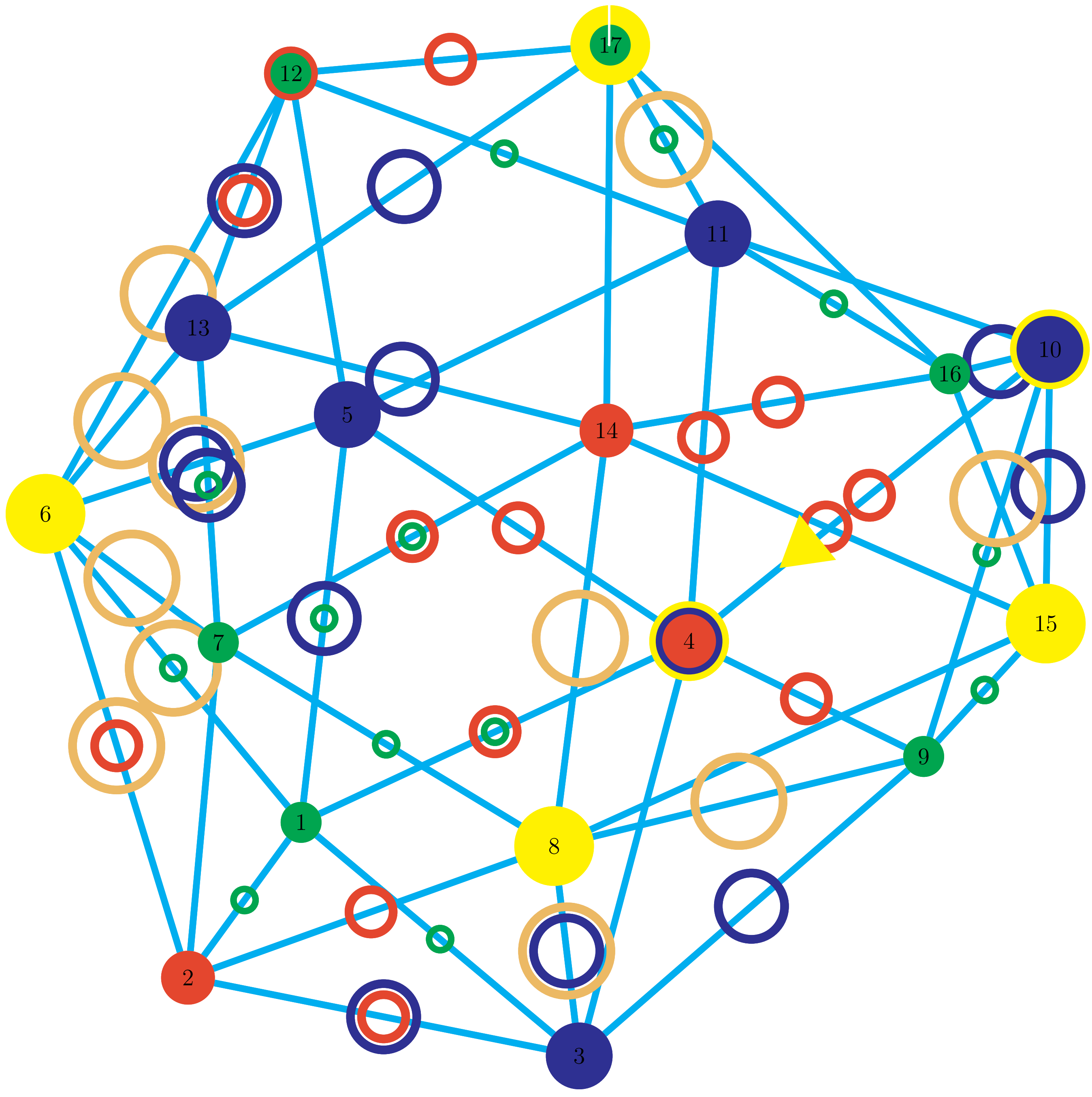

Figure 100. Instructions: 309: place edge 10→4 Yellow ArrowR,

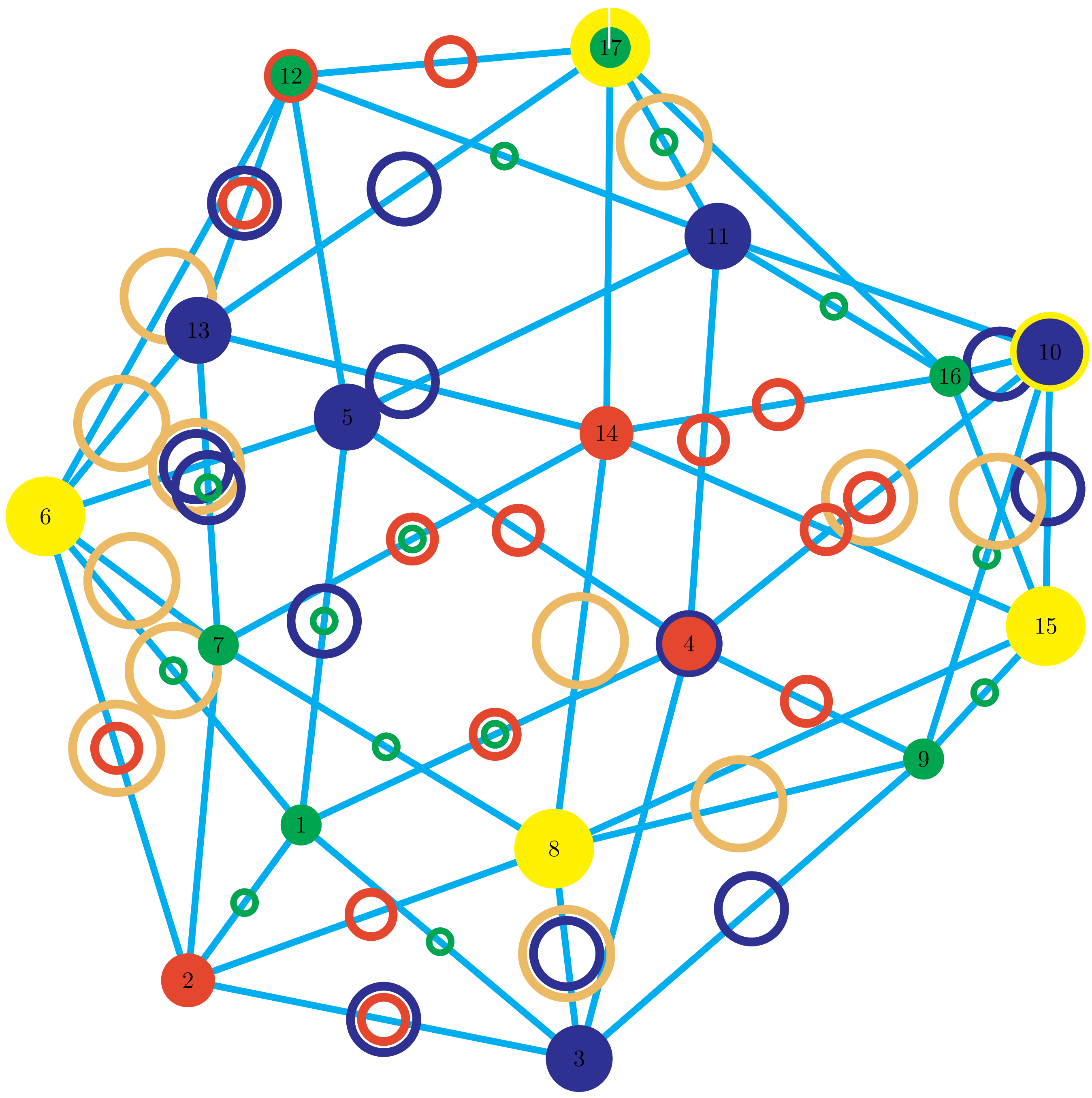

Figure 101. Instructions: 312: color edge 10–4 Yellow, 313: uncolor vertex 4 Yellow,

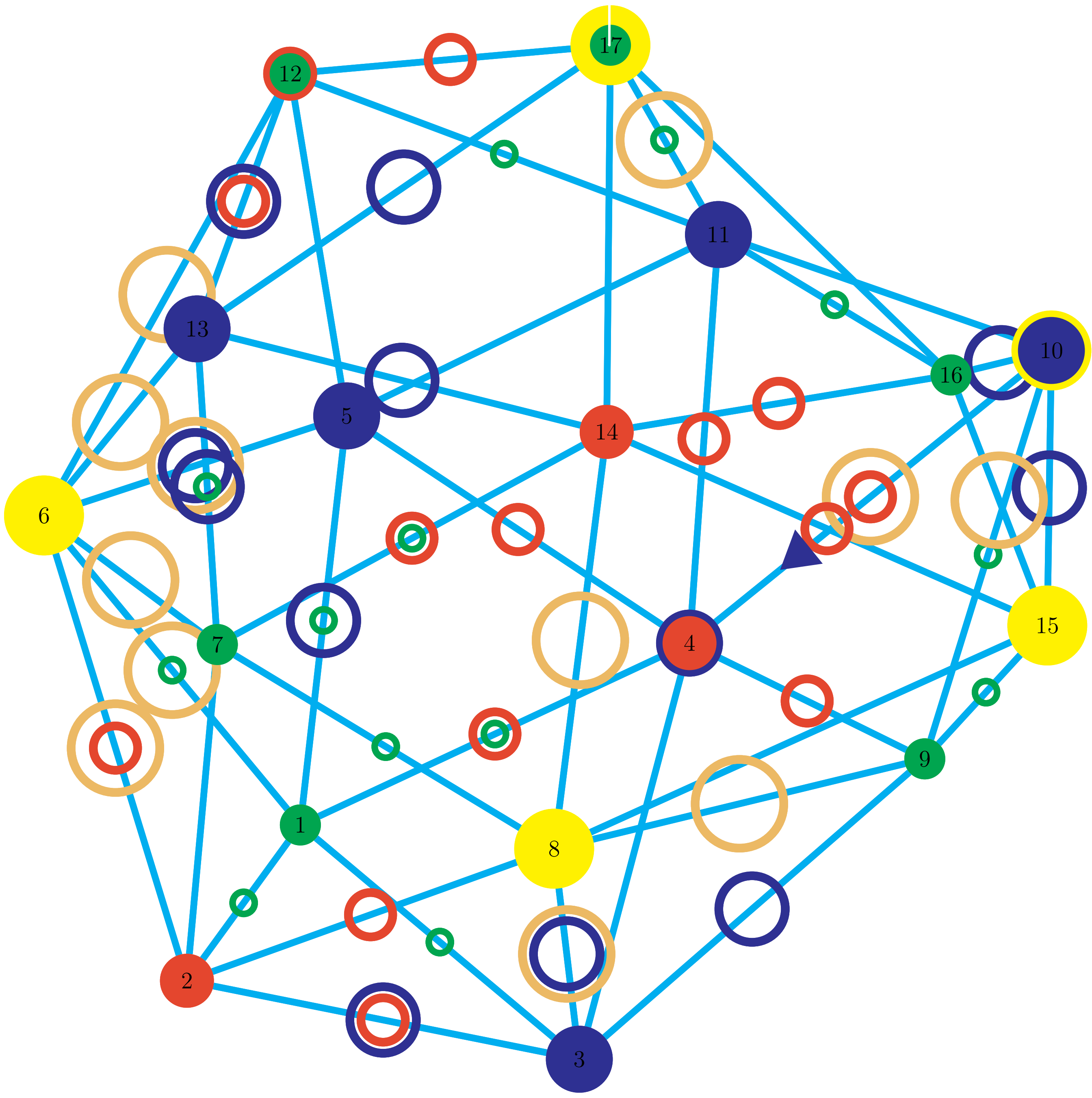

Figure 102. Instructions: 315: place edge 10$\rightarrow$4 Blue ArrowR,

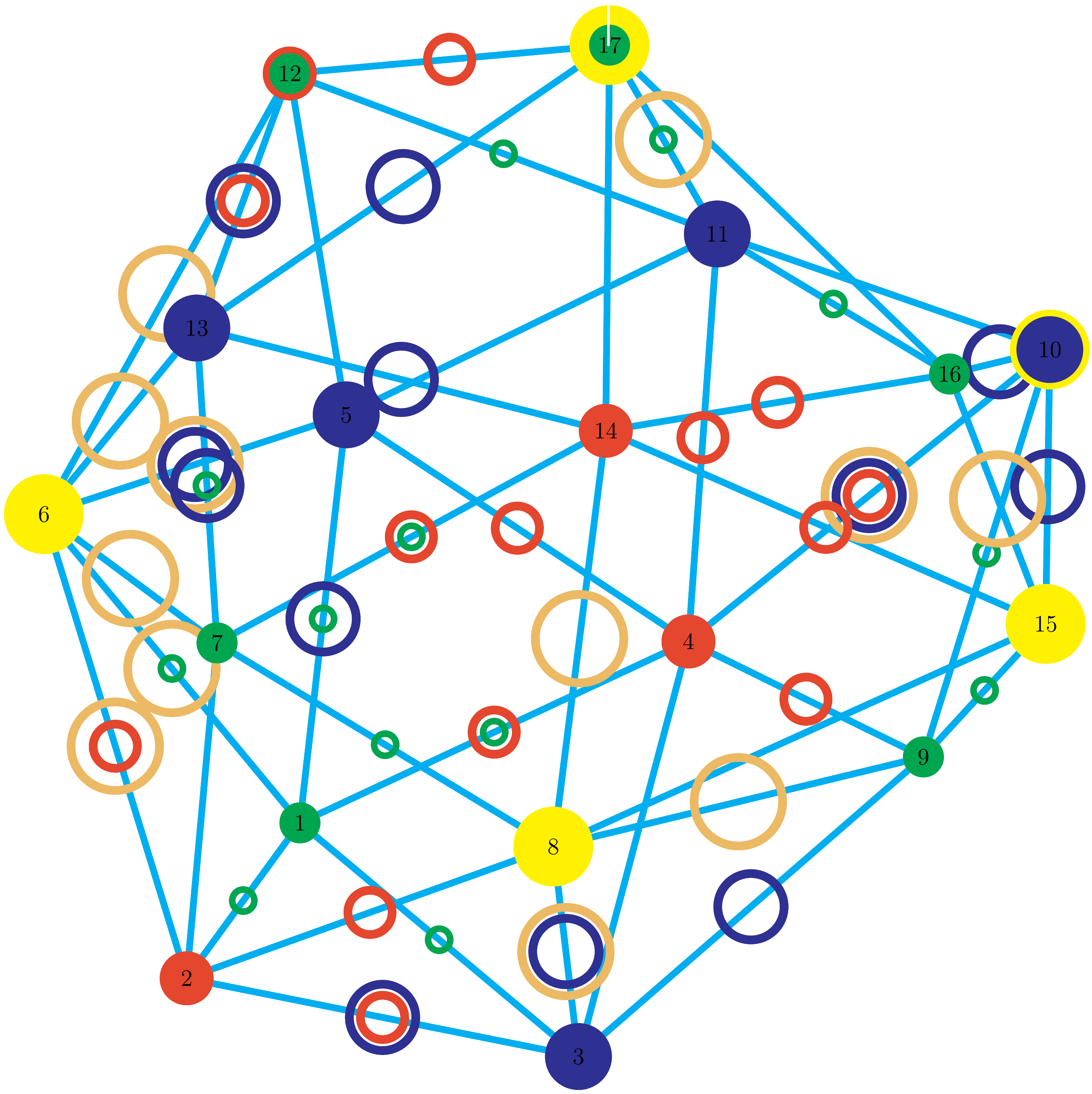

Figure 103. Instructions: 318: color edge 10–4 Blue, 319: uncolor vertex 4 Blue,

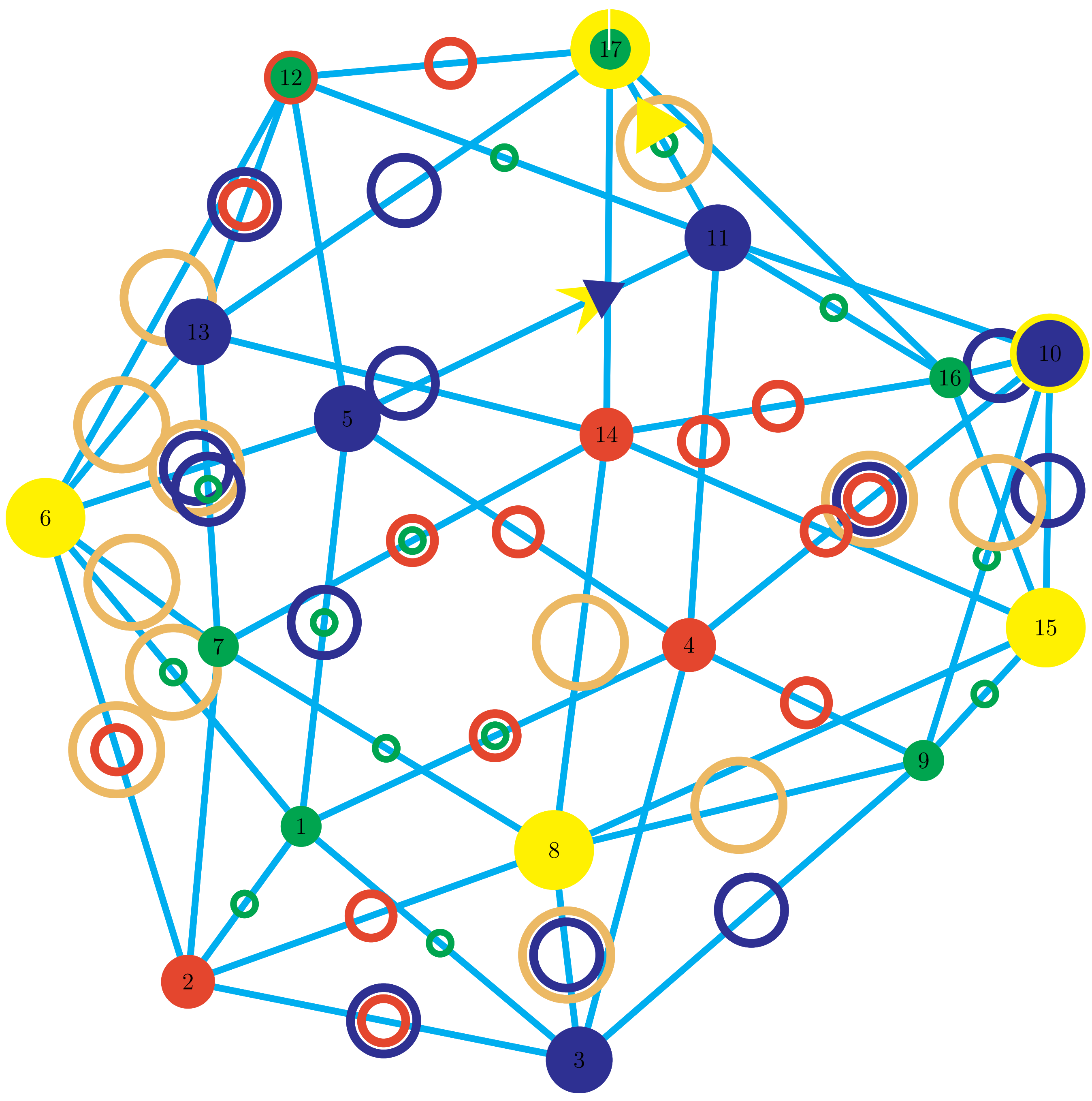

Figure 104. Instructions: 321: place edge 5→11 Blue ArrowR, 322: place edge 5→11 Yellow ArrowI, 323: place edge 11→17 Yellow ArrowR,

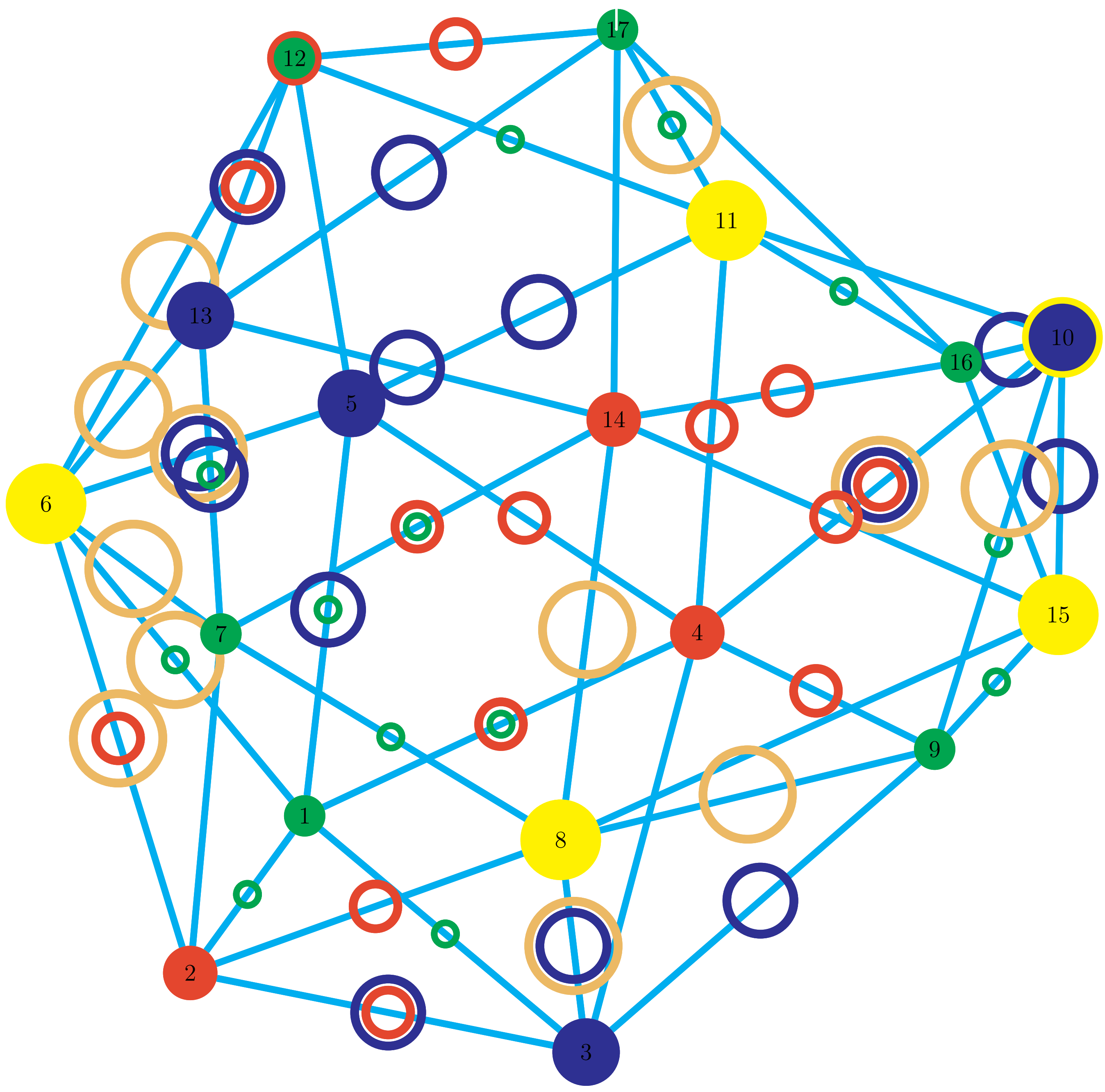

Figure 105. Instructions: 328: color edge 5–11 Blue, 329: uncolor vertex 11 Blue, 330: color vertex 11 Yellow, 331: uncolor vertex 17 Yellow,

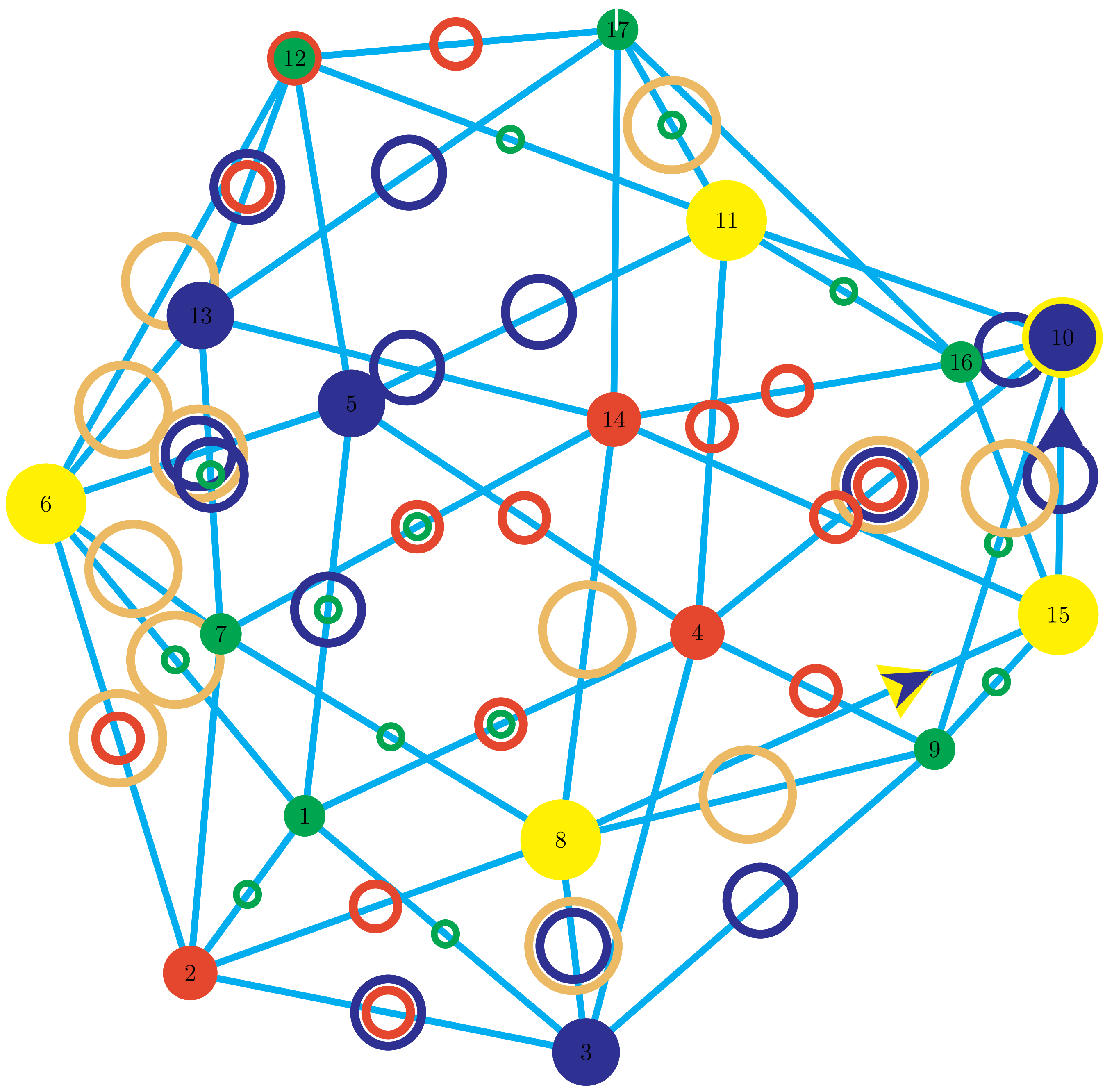

Figure 106. Instructions: 333: place edge 8→15 Yellow ArrowR, 334: place edge 8→15 Blue ArrowI, 335: place edge 15→10 Blue ArrowR,

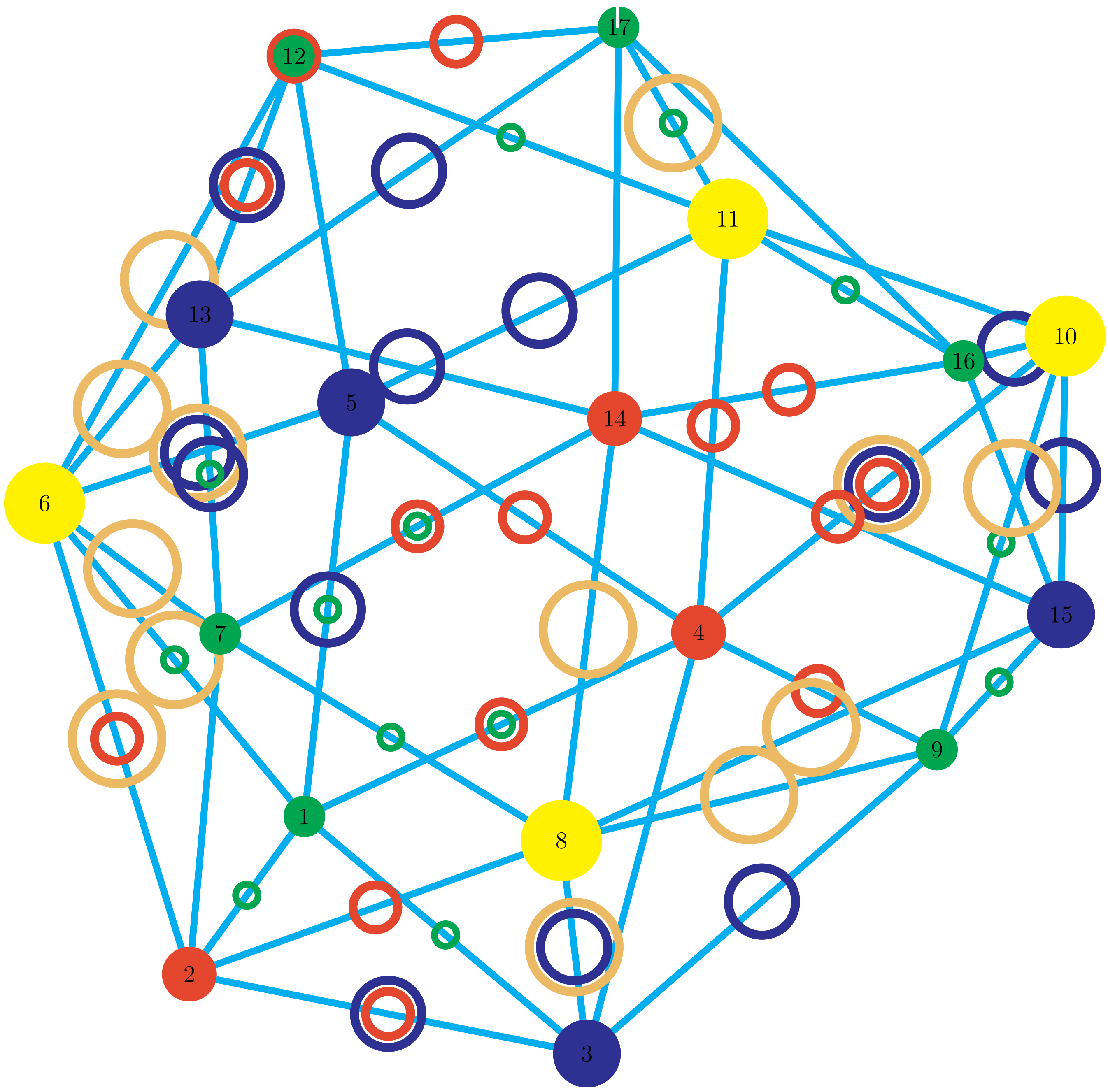

Figure 107. Instructions: 340: color edge 8–15 Yellow, 341: uncolor vertex 15 Yellow, 342: color vertex 15 Blue, 343: uncolor vertex 10 Blue,

Here property B is not satisfied for there is a blue circle on edge 10-16 but no blue checker on either of its vertices. The situation is remedied as follows.

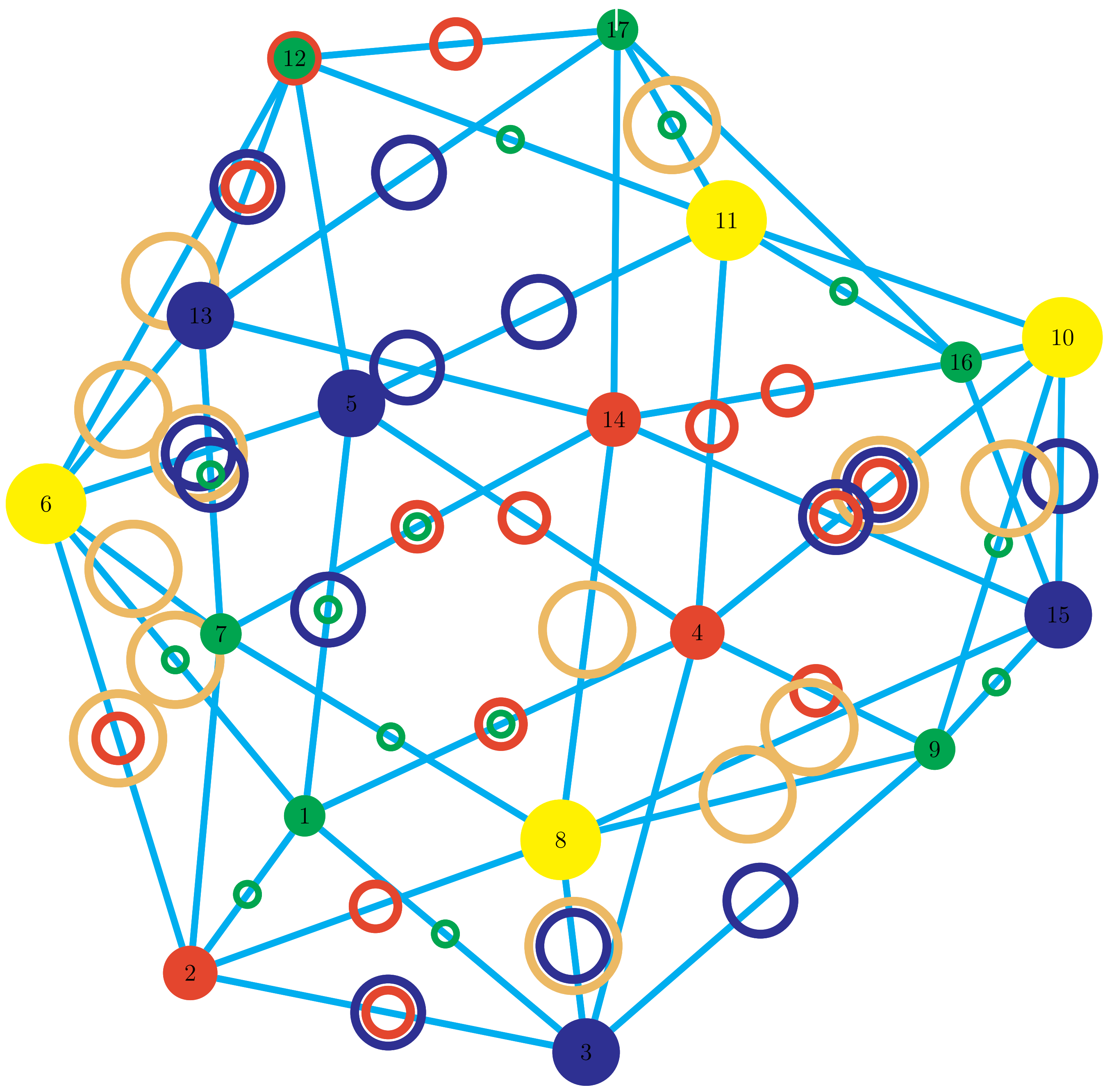

Figure 108. Instructions: 347: color edge 15–14 Blue,

The blue circle on edge 10-16 is flipped to edge 14-15. Similary the Yellow circle on edge 15-16 is flipped to edge 11-16.

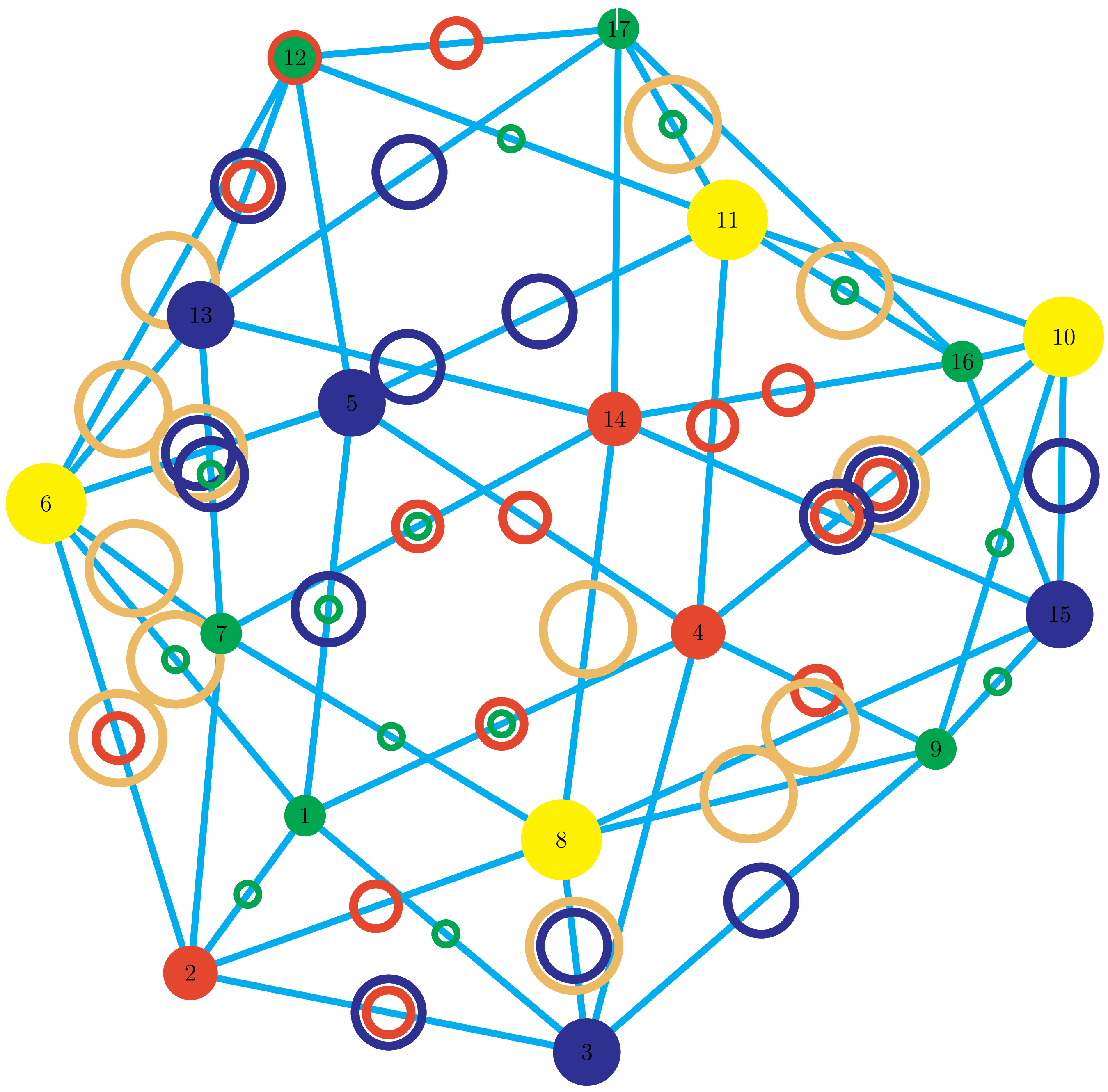

Figure 109. Instructions: 351: color edge 16–11 Yellow,

Similarly the Blue circle on edge 10-4 is flipped to edge 4-5.

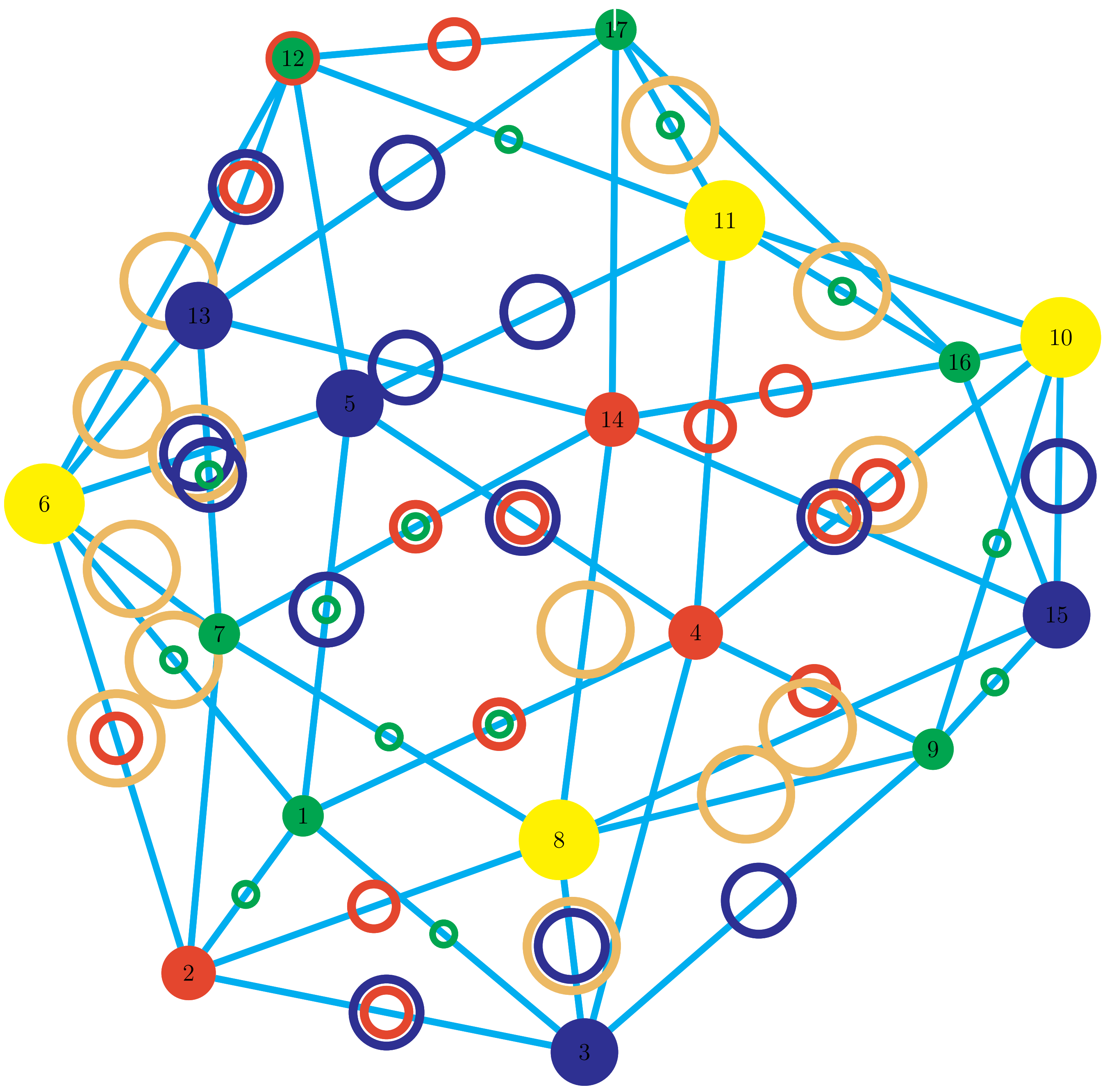

Figure 110. Instructions: 355: color edge 5–4 Blue,

And property B is restored.

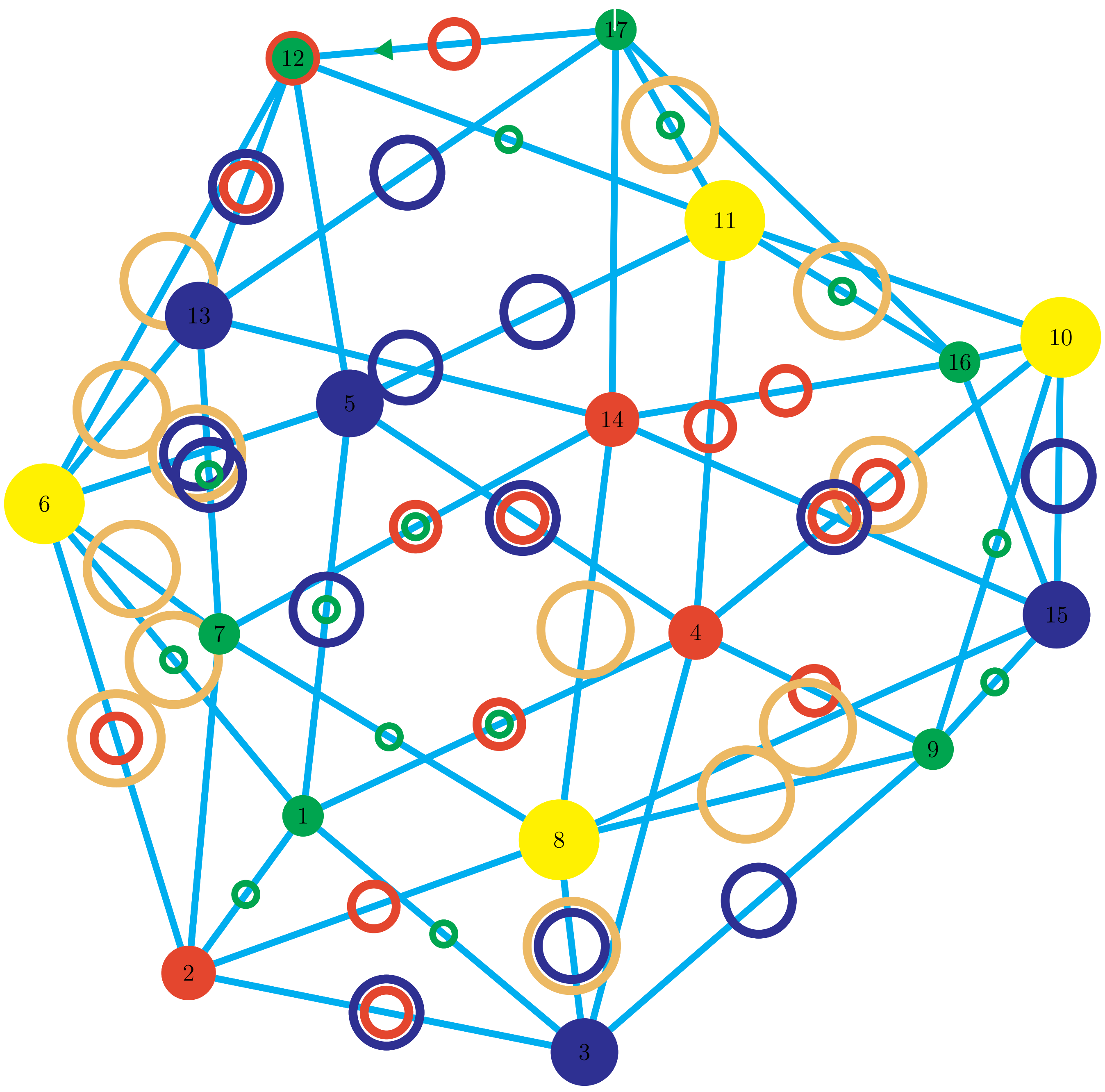

Figure 111. Instructions: 357: place edge 17→12 Green ArrowR,

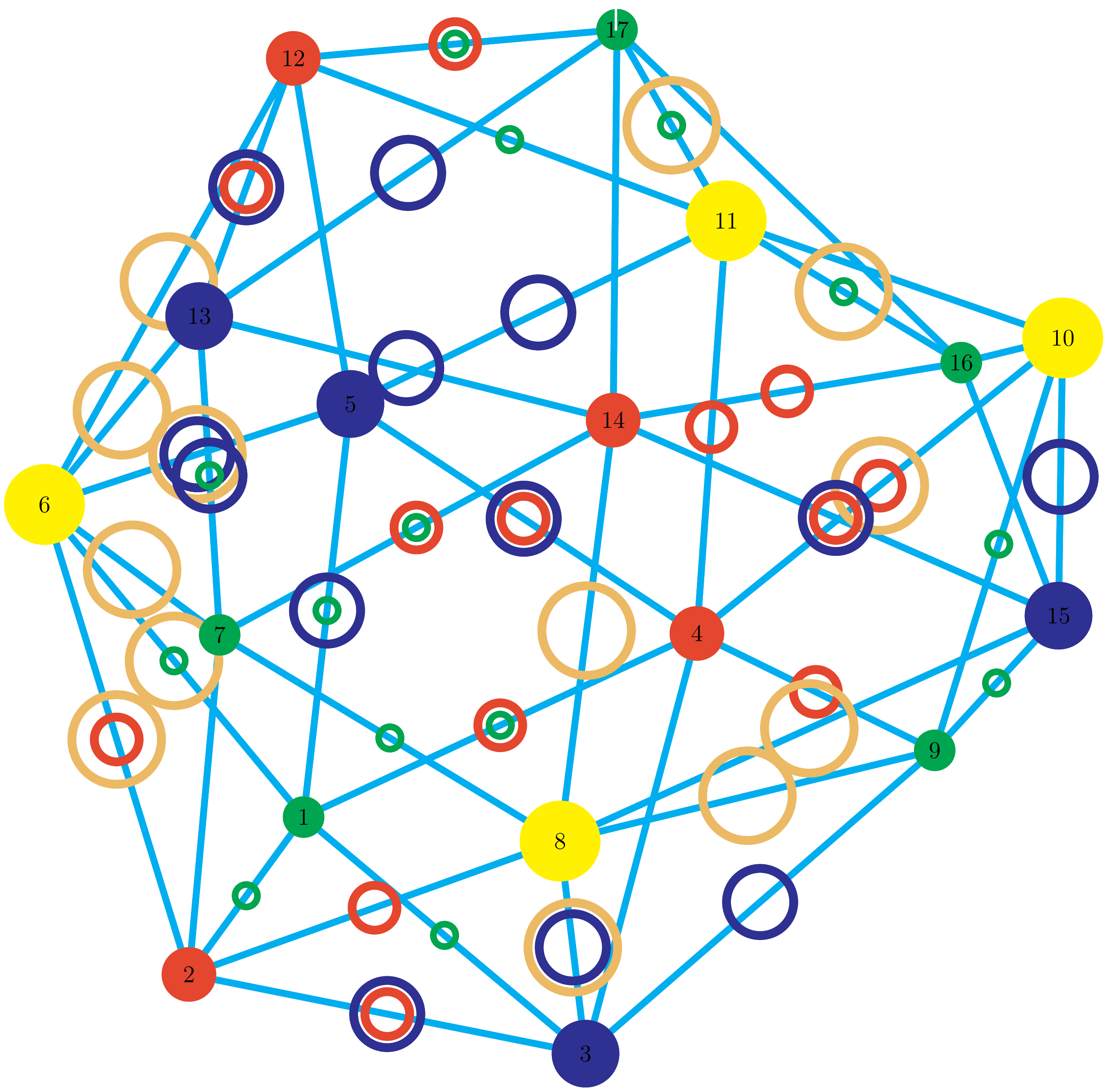

Figure 112. Instructions: 360: color edge 17–12 Green, 361: uncolor vertex 12 Green,

After executing the previous instruction property B fails on edge 11-12. However it is restored by moving the green circle on edge 11-12 to edge 14-17.

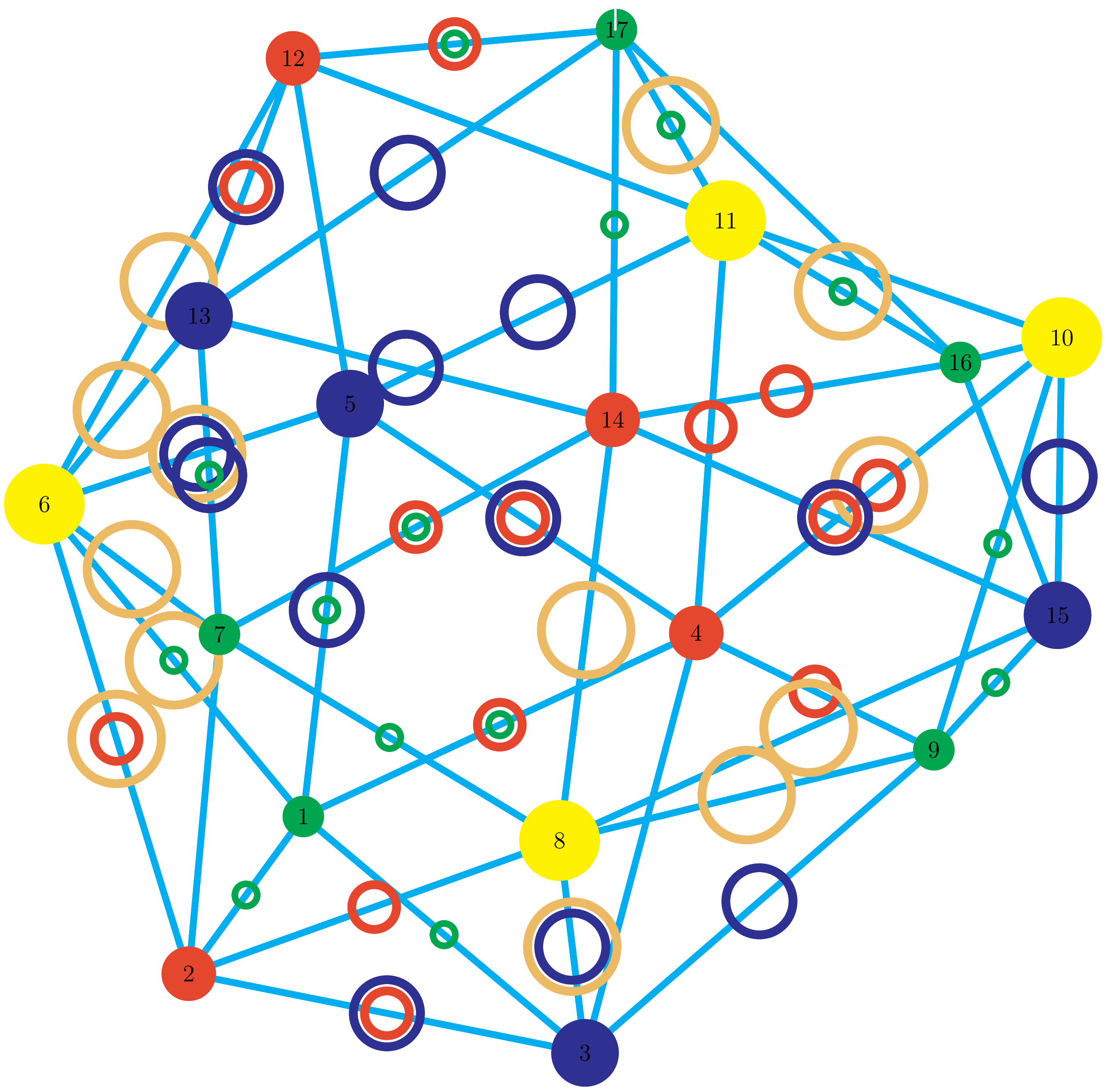

Figure 113. Instructions: 365: color edge 14–17 Green,

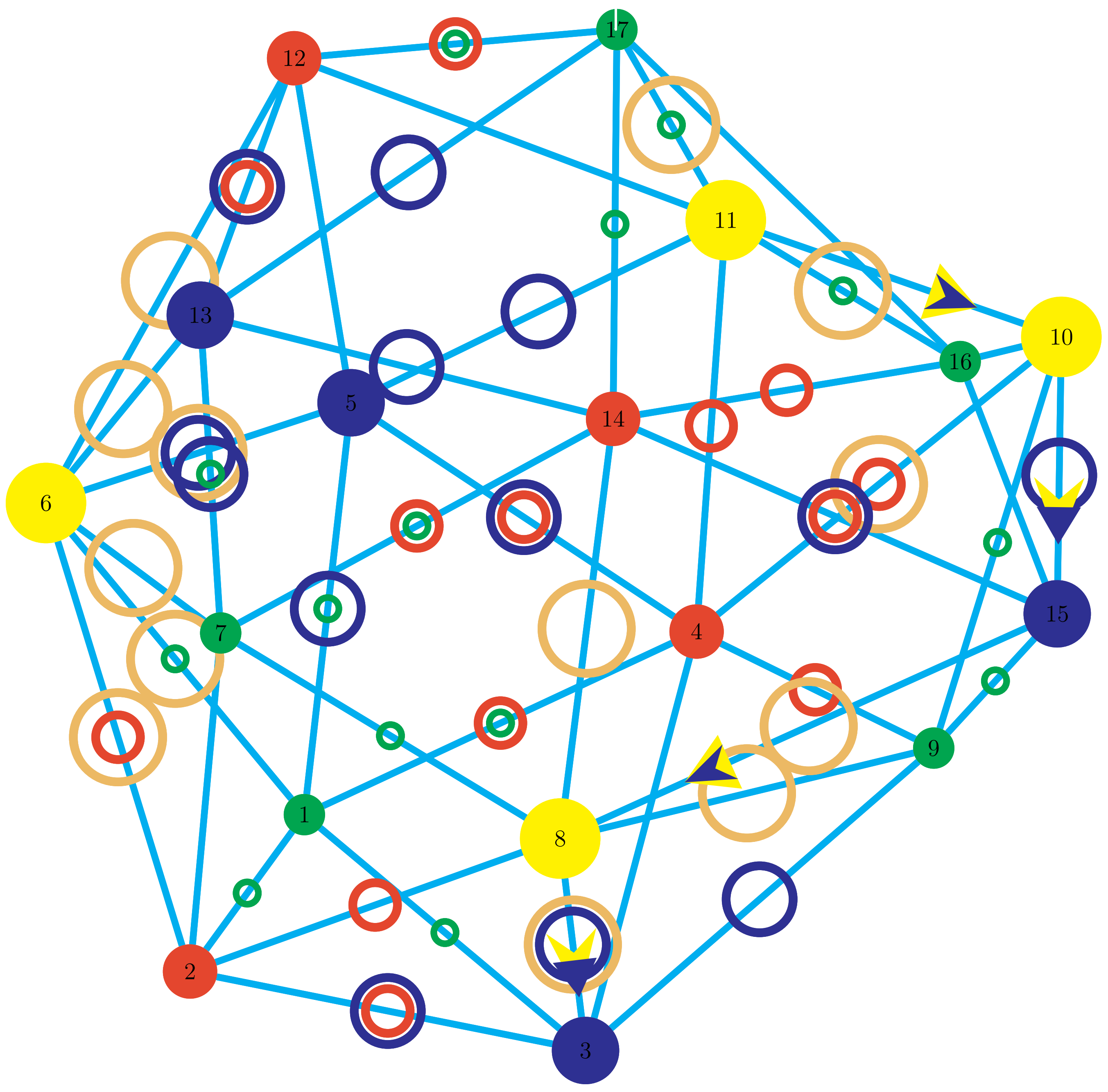

Figure 114. Instructions: 367: place edge 11→10 Yellow ArrowR, 368: place edge 11→10 Blue ArrowI, 369: place edge 10→15 Blue ArrowR, 370: place edge 10→15 Yellow ArrowI, 371: place edge 15→8 Blue ArrowI, 372: place edge 15→8 Yellow ArrowR, 373: place edge 8→3 Blue ArrowR, 374: place edge 8→3 Yellow ArrowI,

Execution of this alternating chain will cause property B to fail. But it is restored in the usual manner.

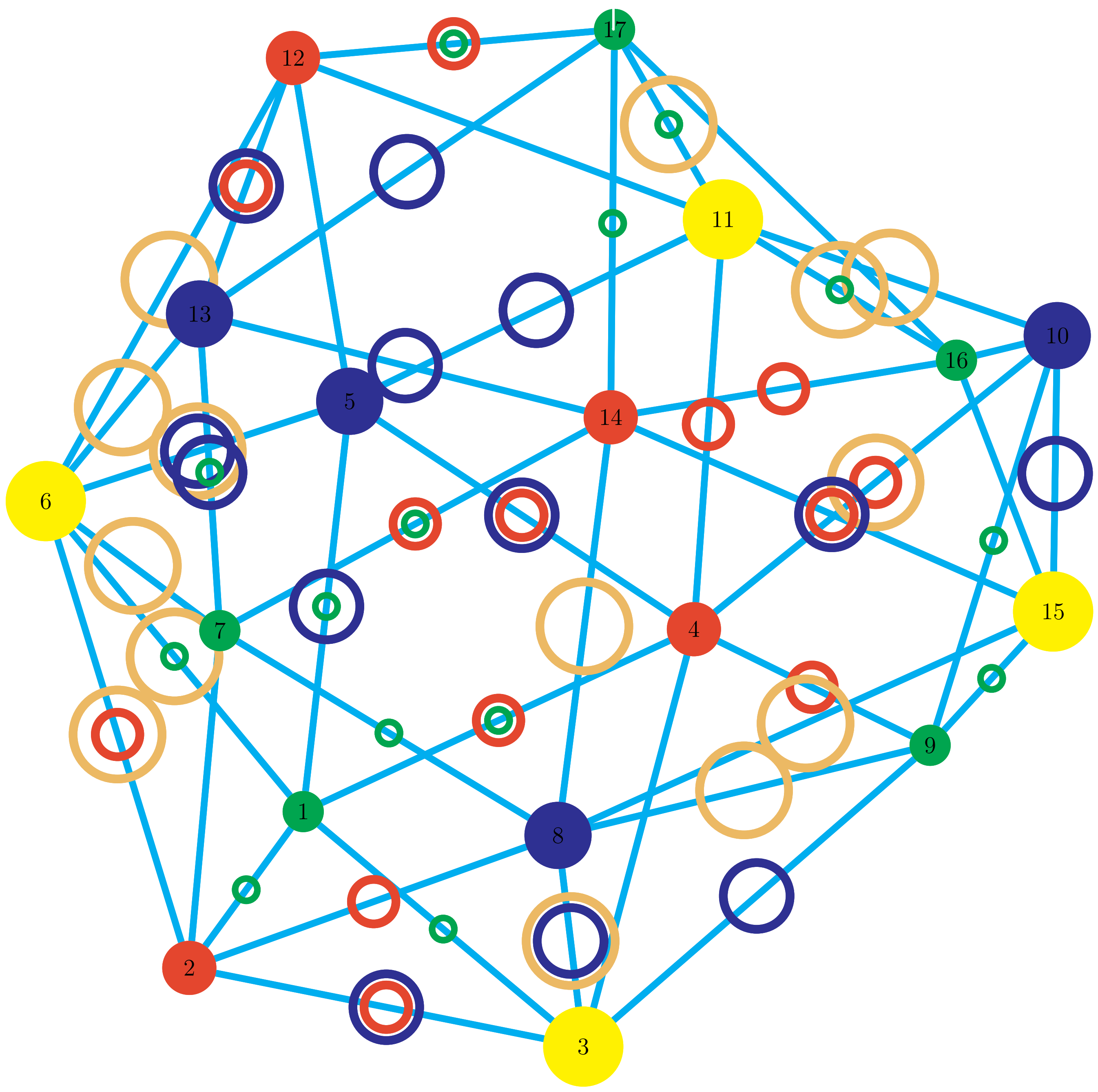

Figure 115. Instructions: 384: color edge 11–10 Yellow, 385: uncolor vertex 10 Yellow, 386: color vertex 10 Blue, 387: uncolor vertex 15 Blue, 388: color vertex 15 Yellow, 389: uncolor vertex 8 Yellow, 390: color vertex 8 Blue, 391: uncolor vertex 3 Blue, 392: color vertex 3 Yellow,

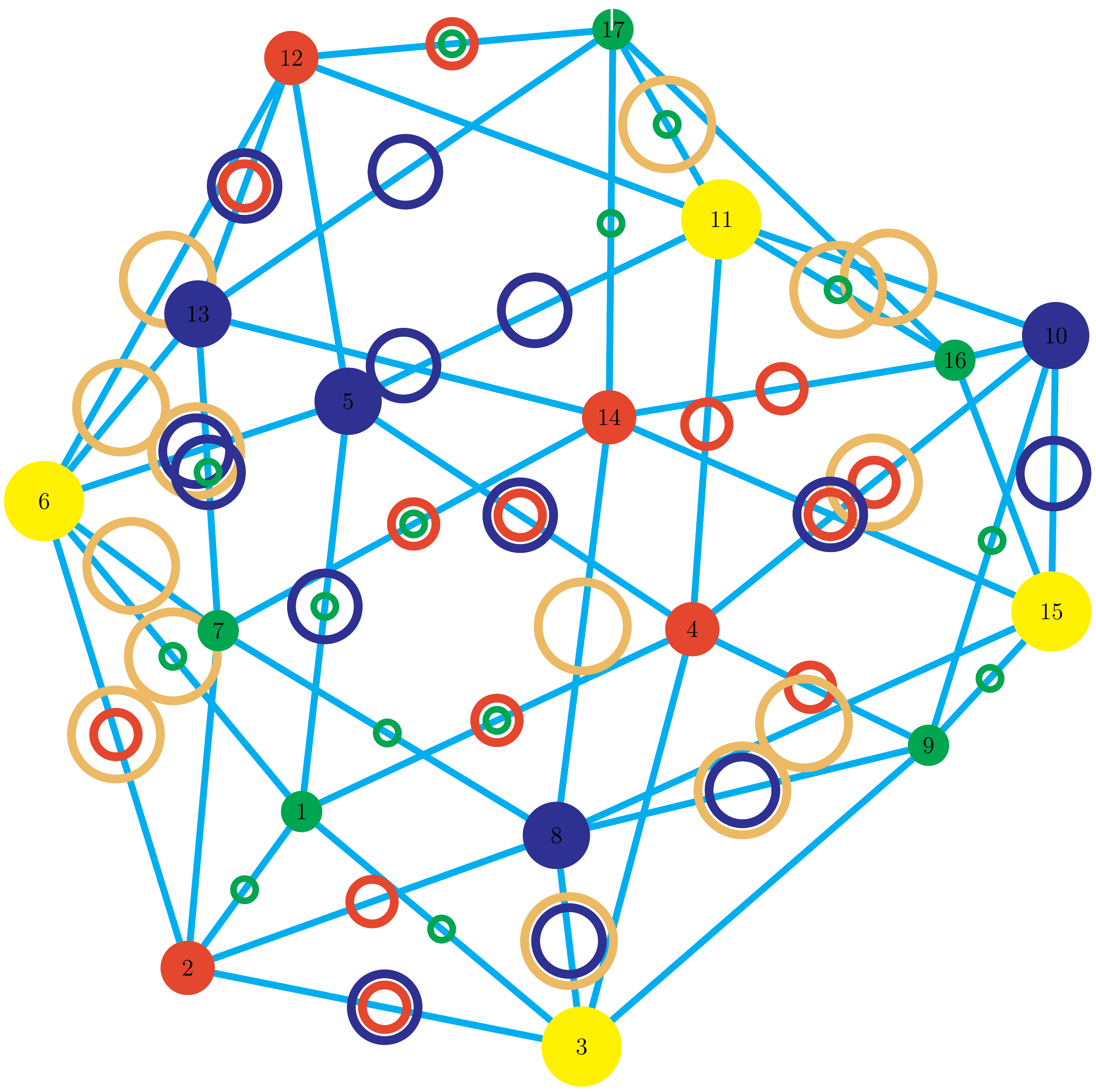

Figure 116. Instructions: 396: color edge 9–8 Blue,

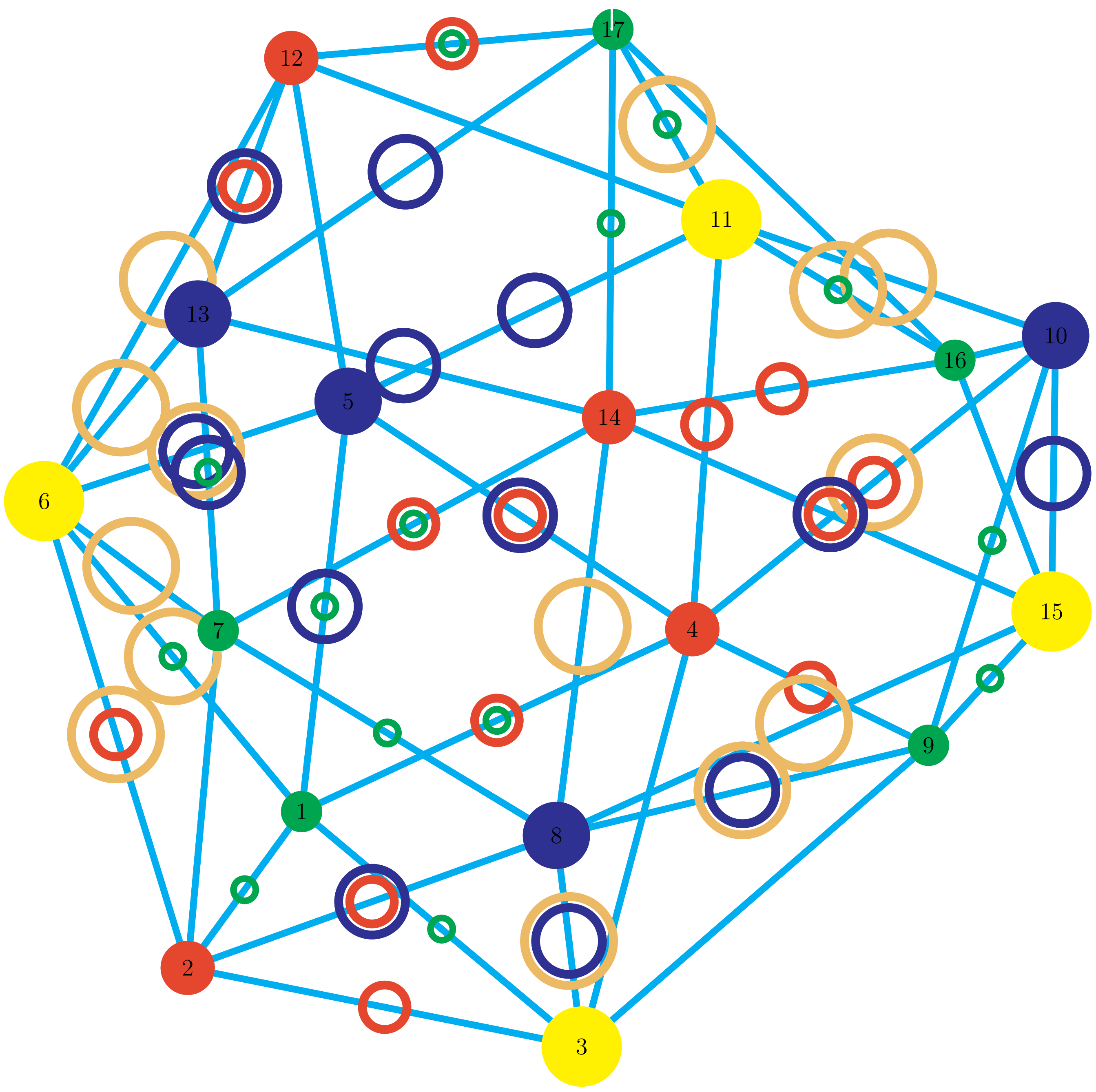

Figure 117. Instructions: 400: color edge 2–8 Blue,

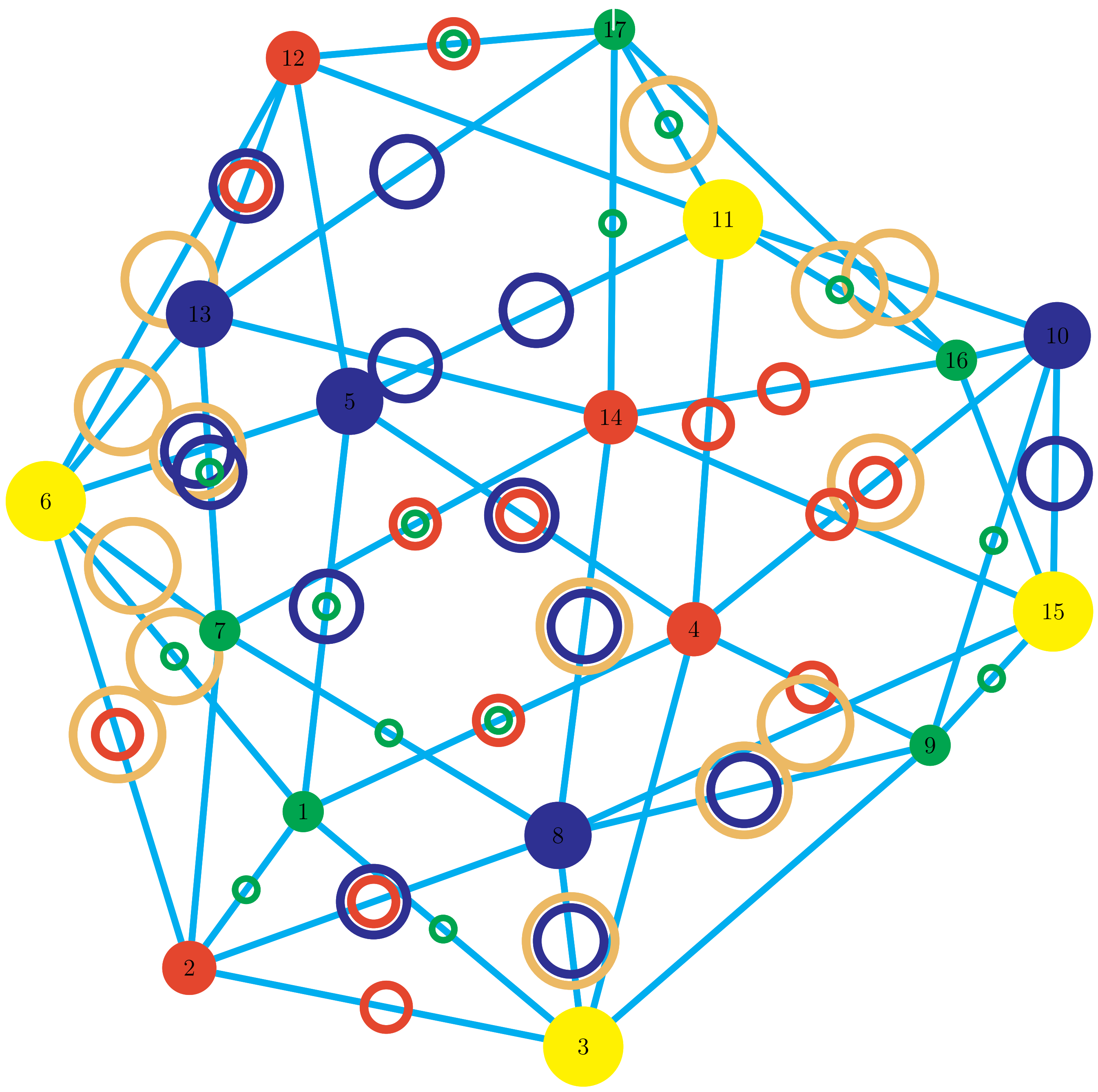

Figure 118. Instructions: 404: color edge 8–14 Blue,

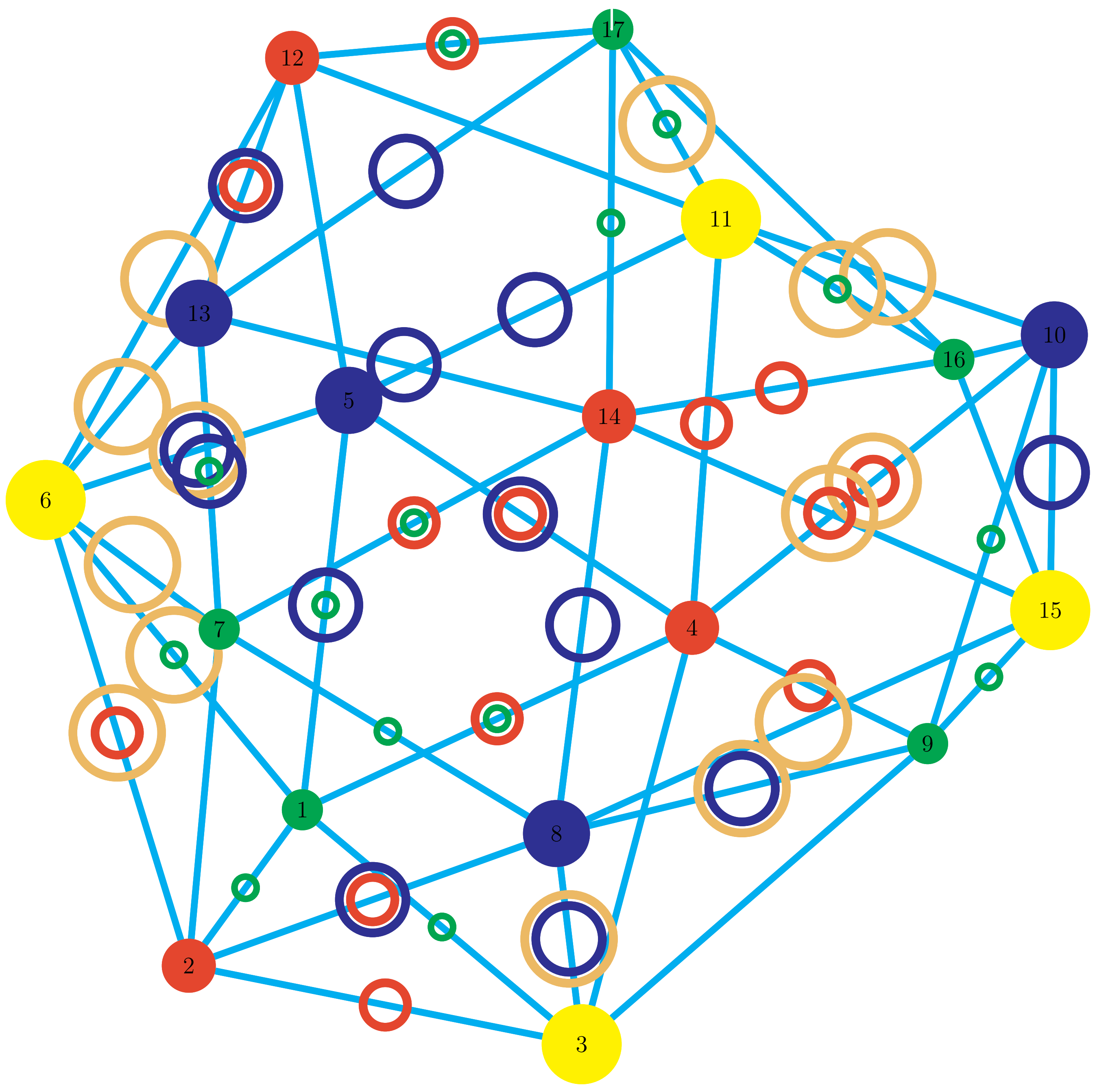

Figure 119. Instructions: 408: color edge 15–14 Yellow,

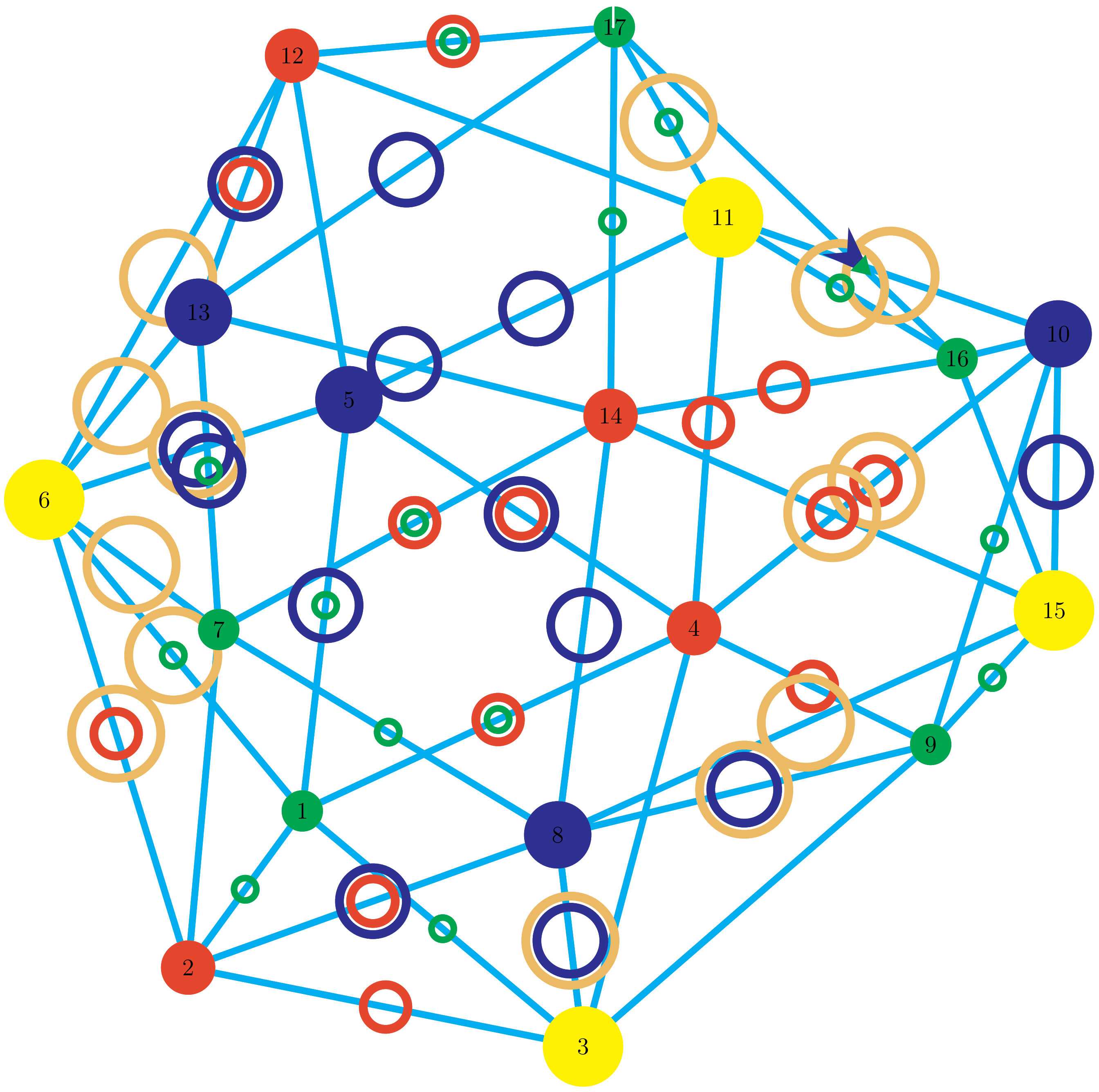

Figure 120. Instructions: 410: place edge 17→16 Green ArrowR, 411: place edge 17→16 Blue ArrowI,

Execution of this alternating chain will cause property B to fail. But it is restored in the usual manner.

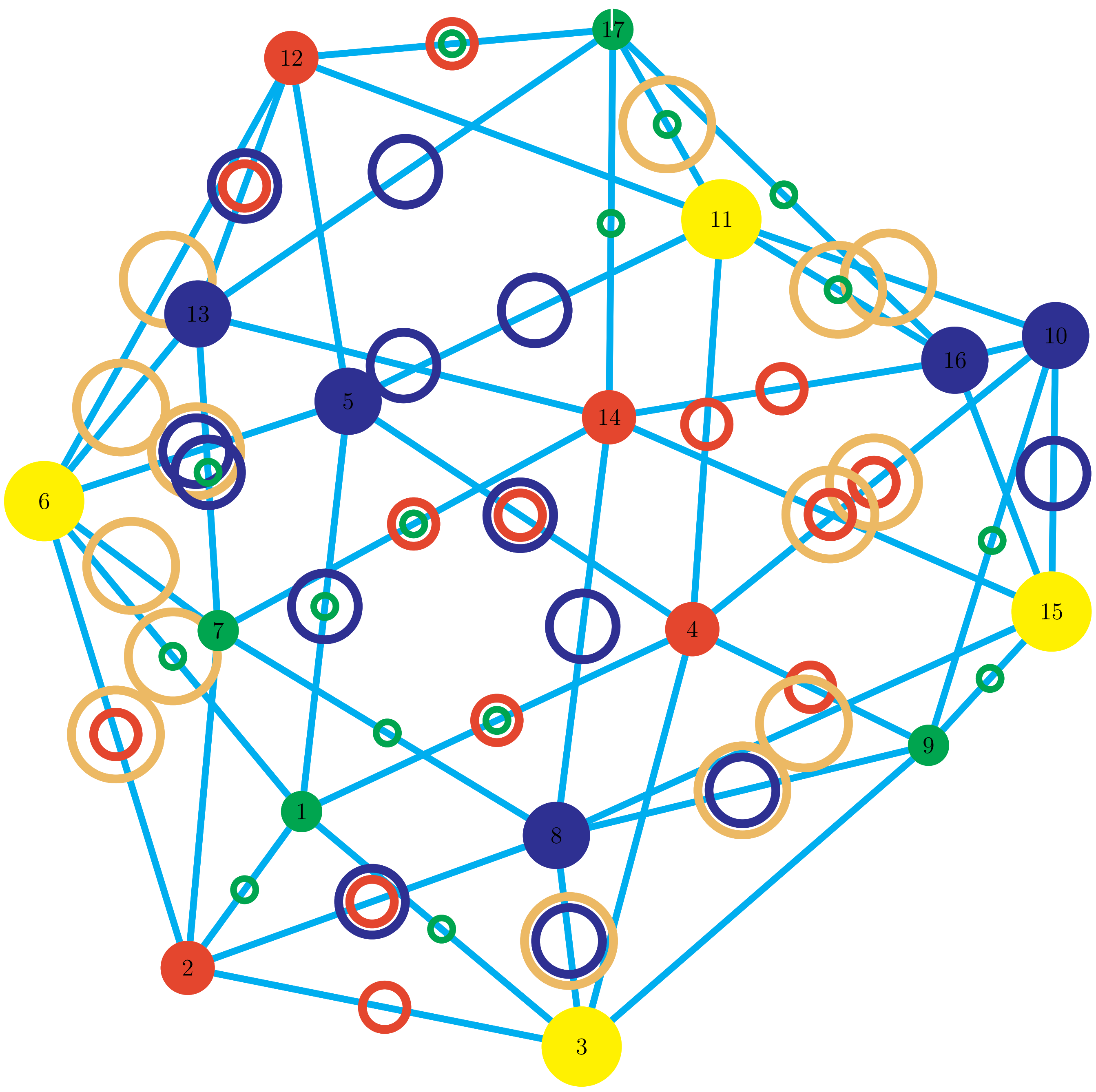

Figure 121. Instructions: 415: color edge 17–16 Green, 416: uncolor vertex 16 Green, 417: color vertex 16 Blue,

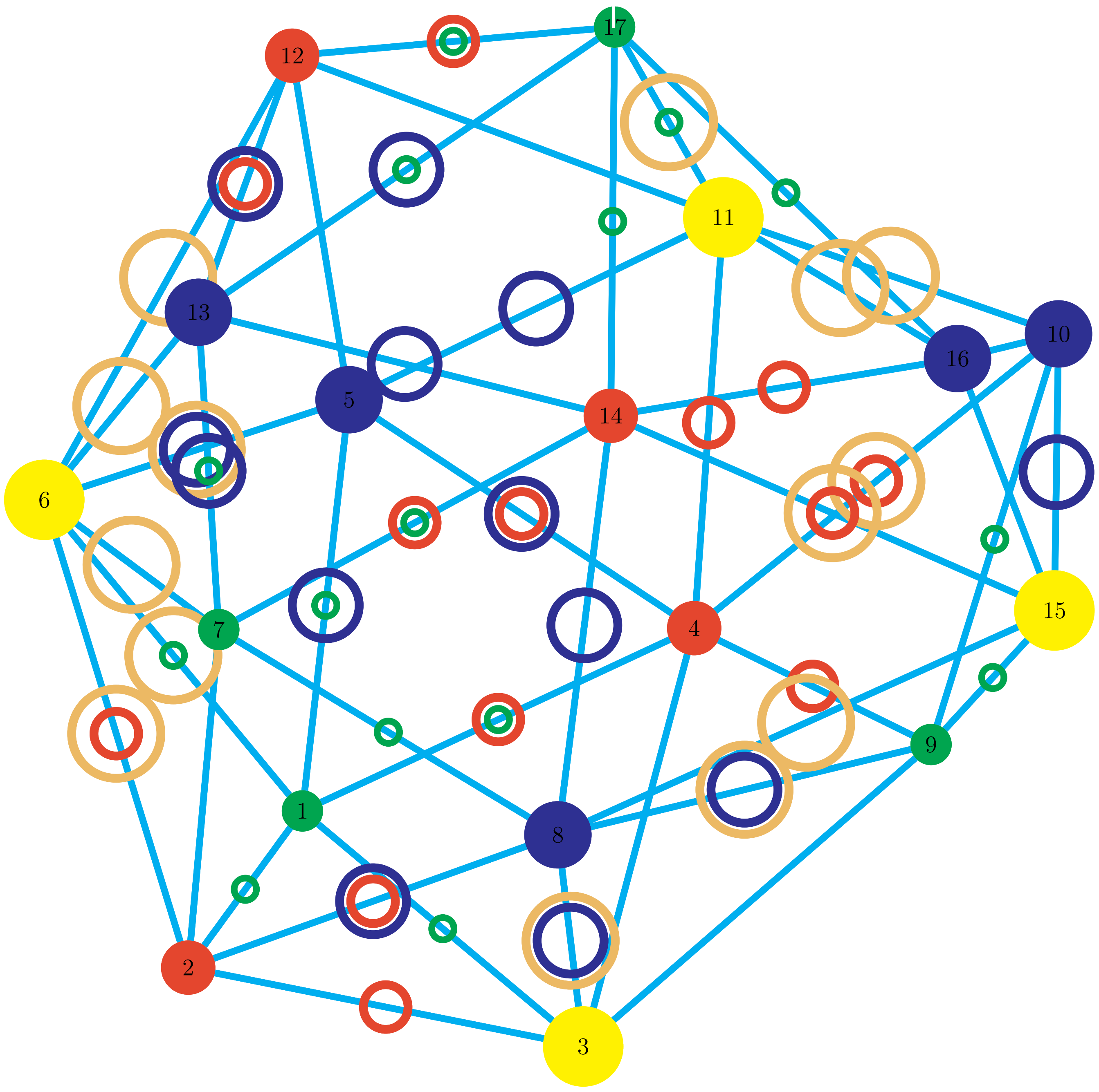

Figure 122. Instructions: 421: color edge 17–13 Green,

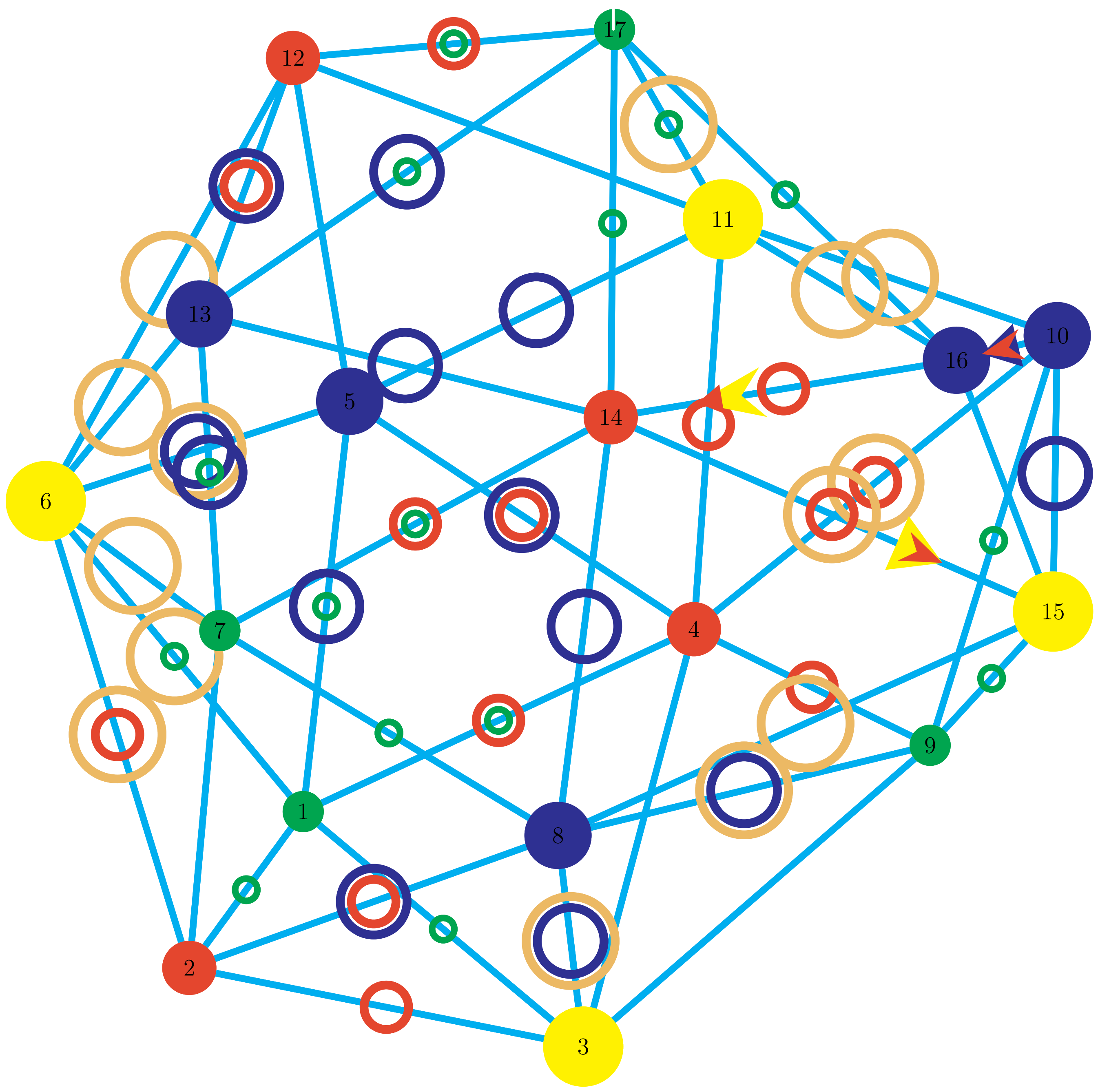

Figure 123. Instructions: 423: place edge 10→16 Blue ArrowR, 424: place edge 10→16 Red ArrowI, 425: place edge 16→14 Red ArrowR, 426: place edge 16→14 Yellow ArrowI, 427: place edge 14→15 Red ArrowI, 428: place edge 14→15 Yellow ArrowR,

Execution of this alternating chain will cause property B to fail. But it is restored in the usual manner.

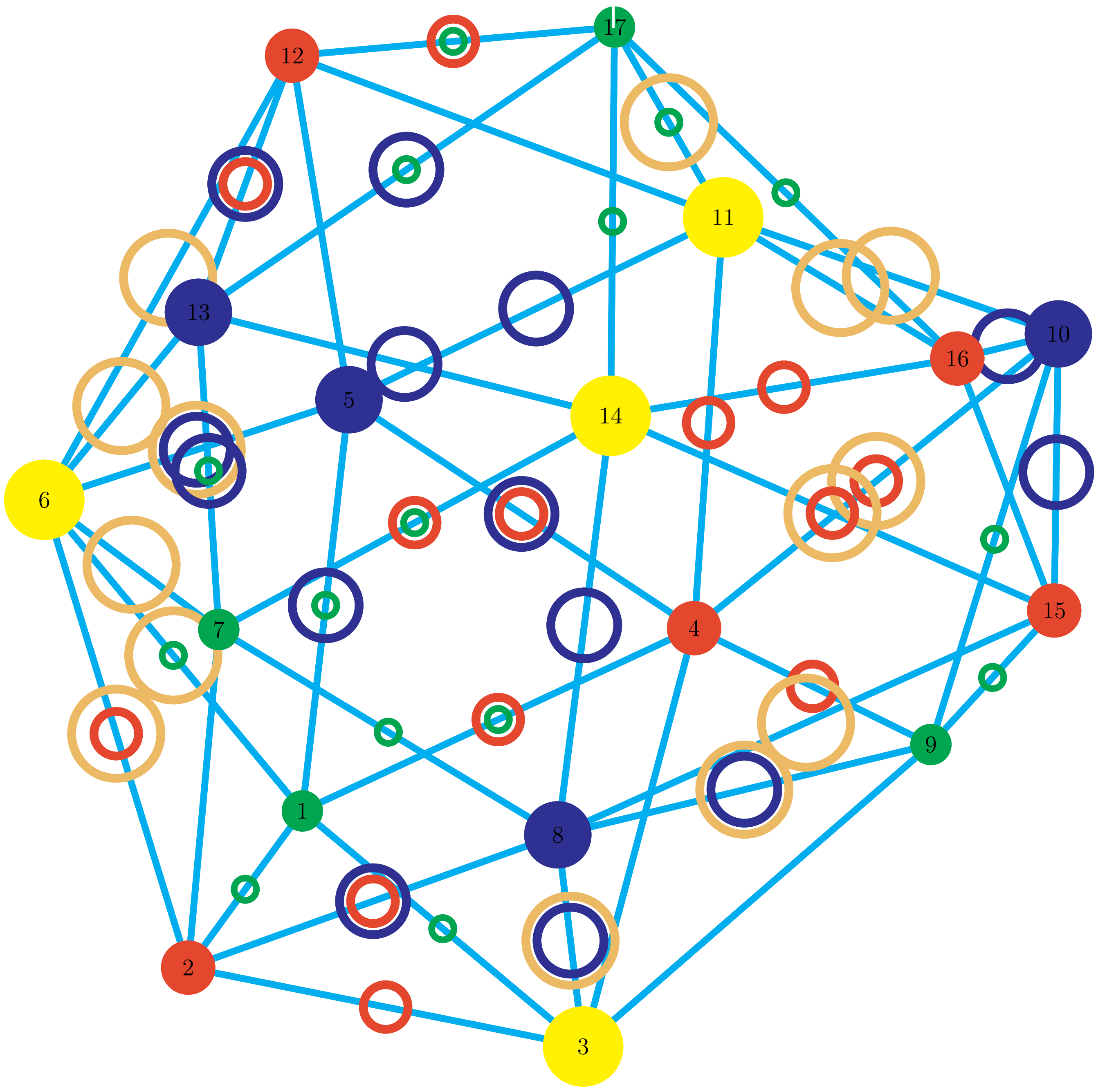

Figure 124. Instructions: 436: color edge 10–16 Blue, 437: uncolor vertex 16 Blue, 438: color vertex 16 Red, 439: uncolor vertex 14 Red, 440: color vertex 14 Yellow, 441: uncolor vertex 15 Yellow, 442: color vertex 15 Red,

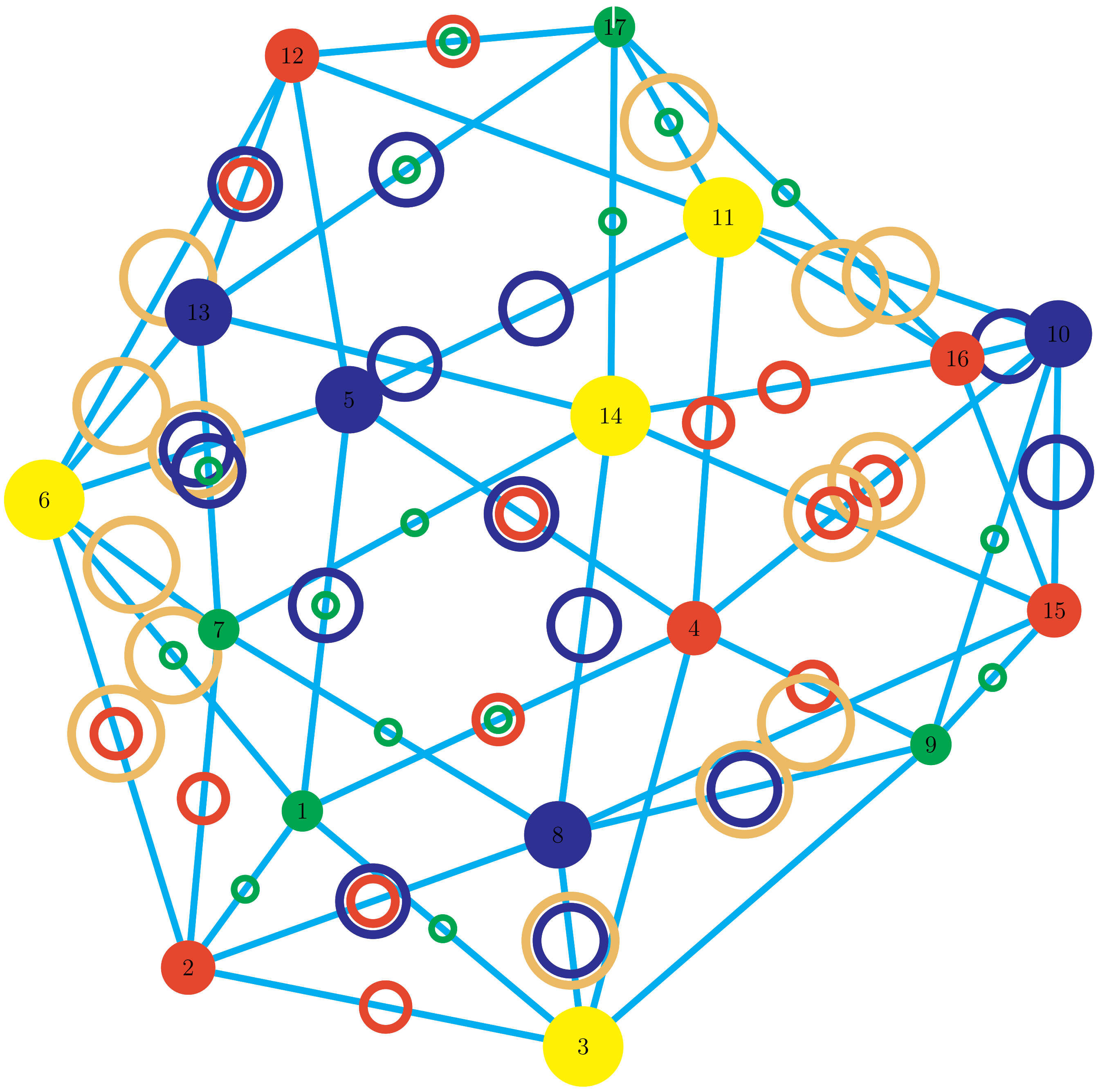

Figure 125. Instructions: 446: color edge 7–2 Red,

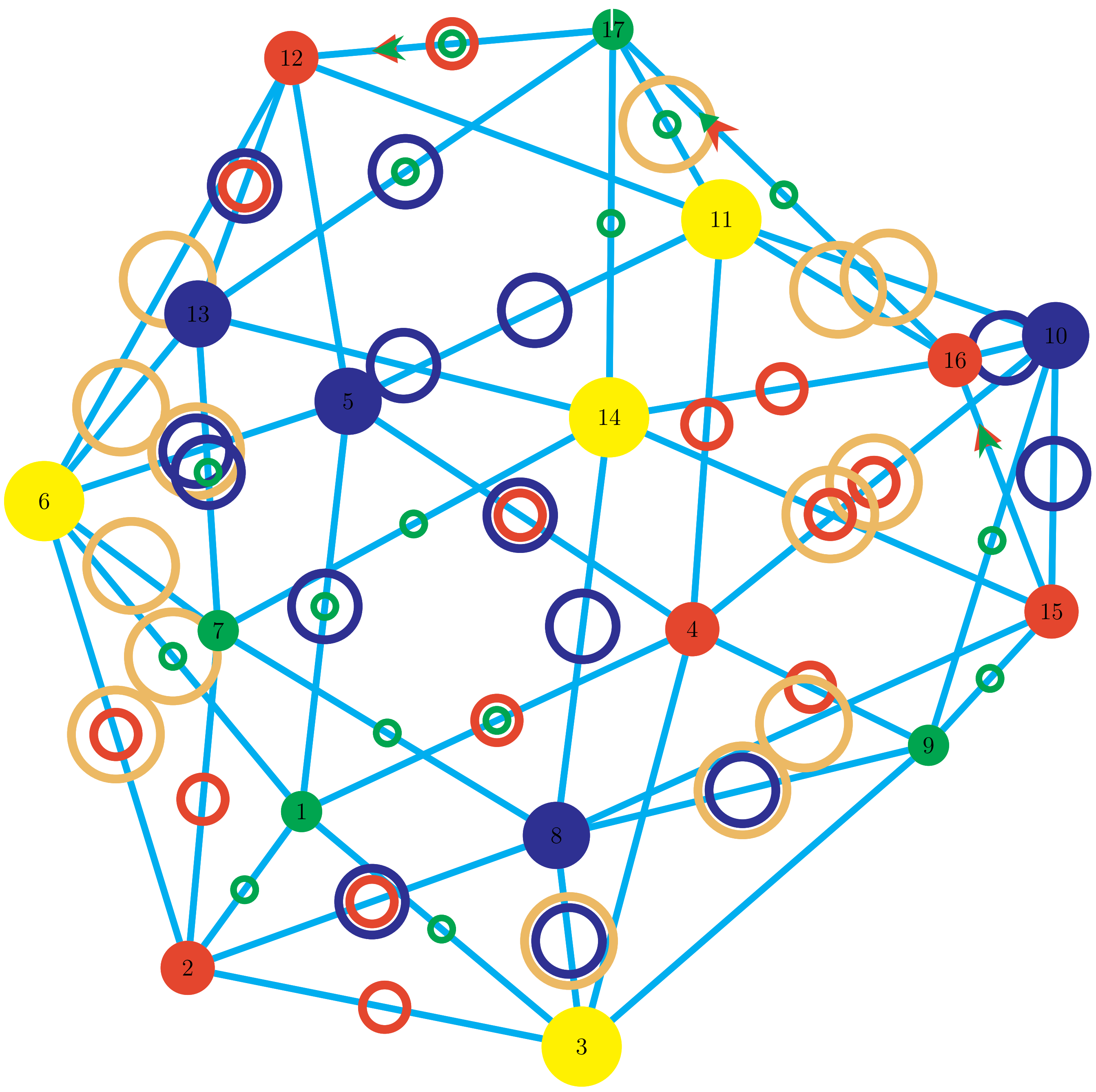

Figure 126. Instructions: 448: place edge 15→16 Red ArrowR, 449: place edge 15→16 Green ArrowI, 450: place edge 16→17 Green ArrowR, 451: place edge 16→17 Red ArrowI, 452: place edge 17→12 Green ArrowI, 453: place edge 17→12 Red ArrowR,

Execution of this alternating chain will cause property B to fail. But it is restored in the usual manner.

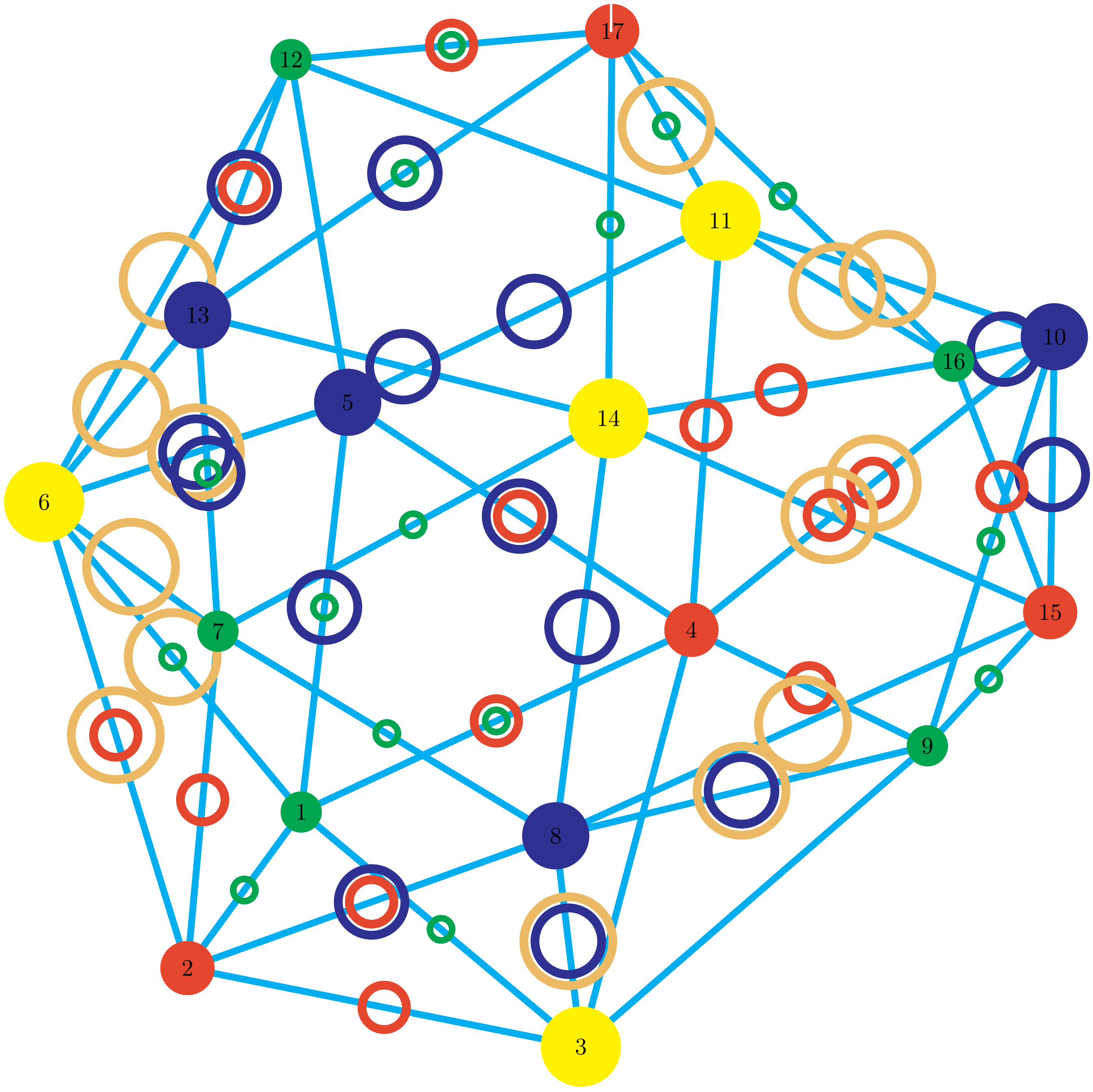

Figure 127. Instructions: 461: color edge 15–16 Red, 462: uncolor vertex 16 Red, 463: color vertex 16 Green, 464: uncolor vertex 17 Green, 465: color vertex 17 Red, 466: uncolor vertex 12 Red, 467: color vertex 12 Green,

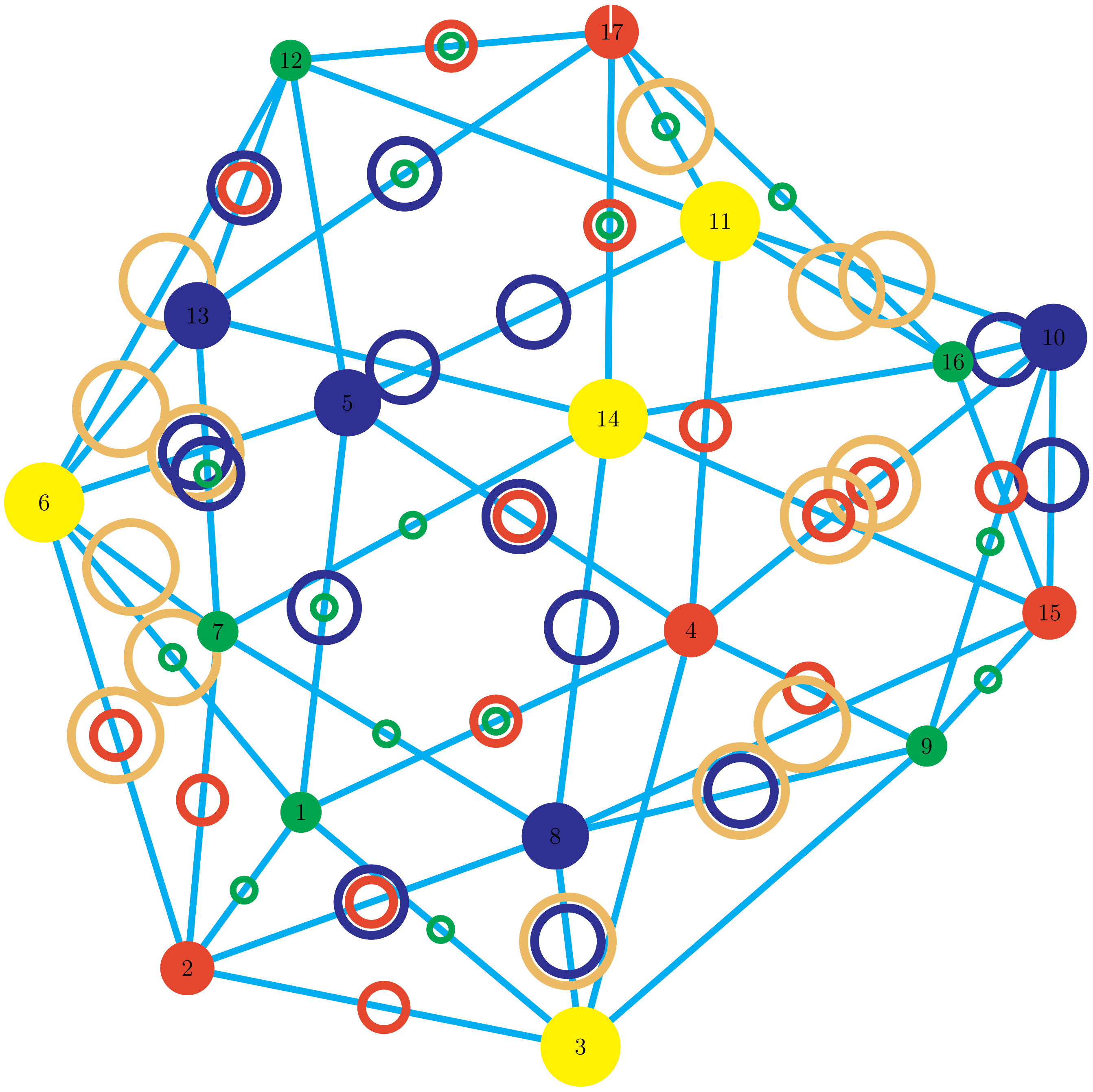

Figure 128. Instructions: 471: color edge 17–14 Red,

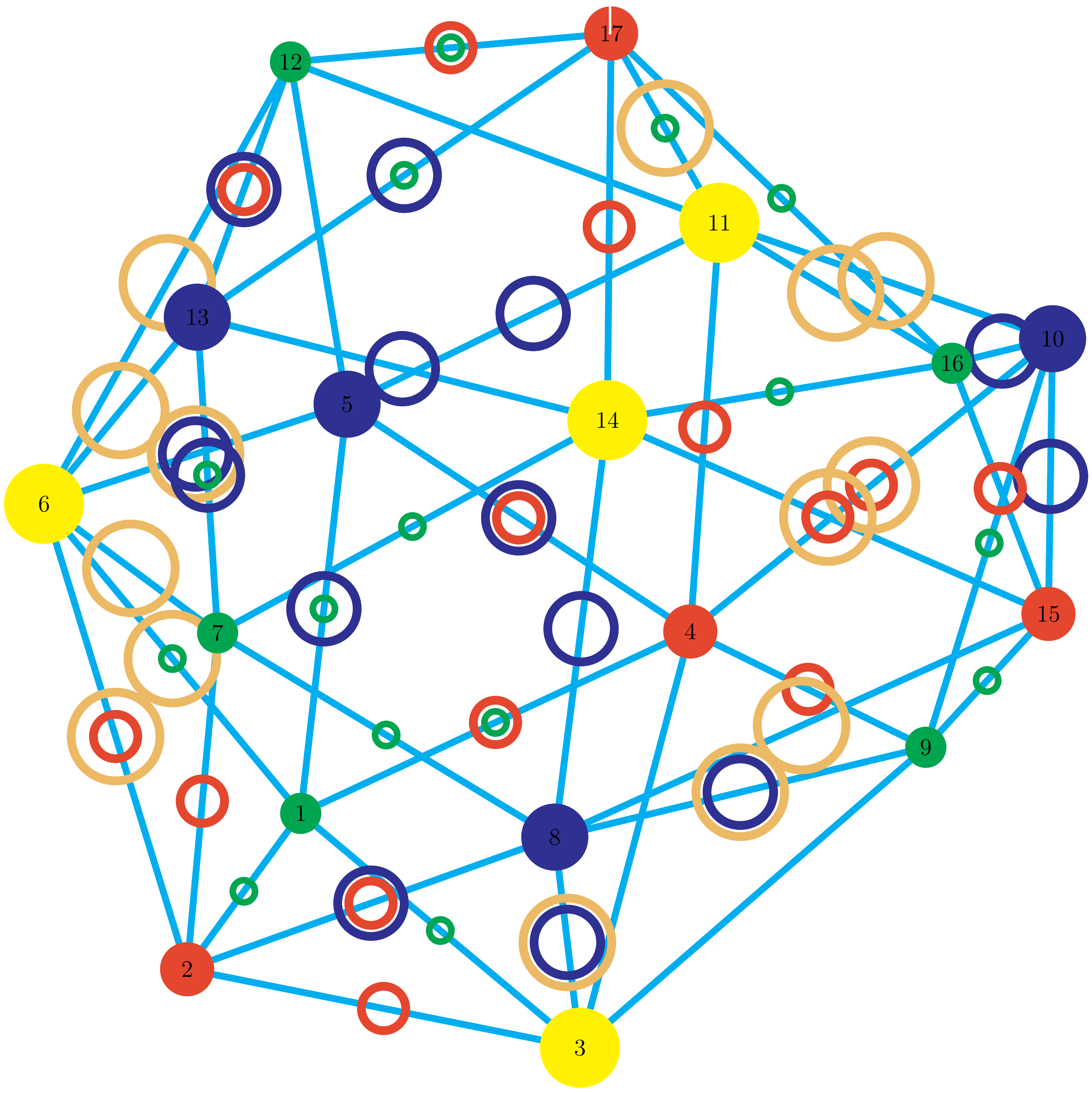

Figure 129. Instructions: 475: color edge 16–14 Green,

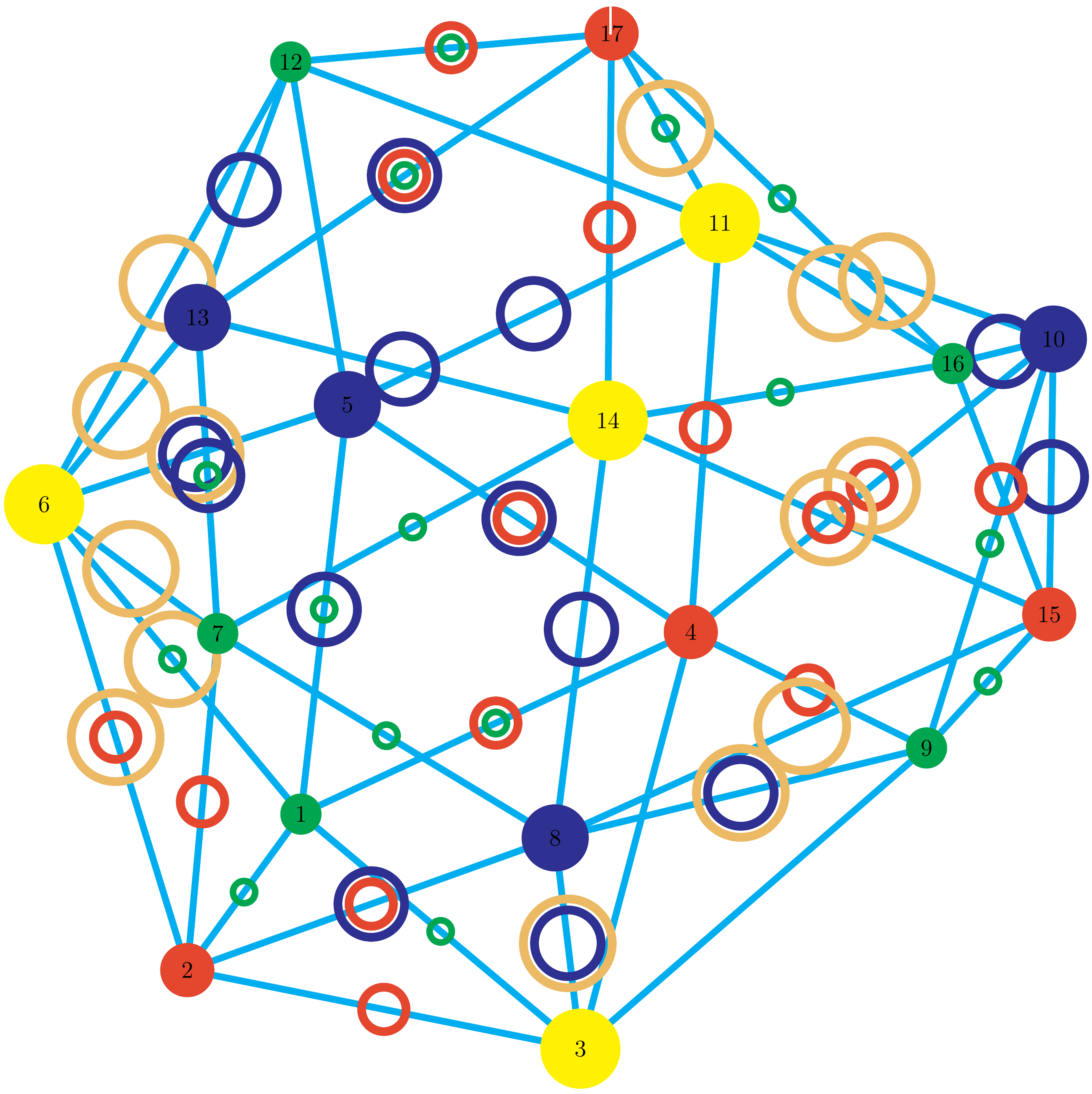

Figure 130. Instructions: 479: color edge 17–13 Red,

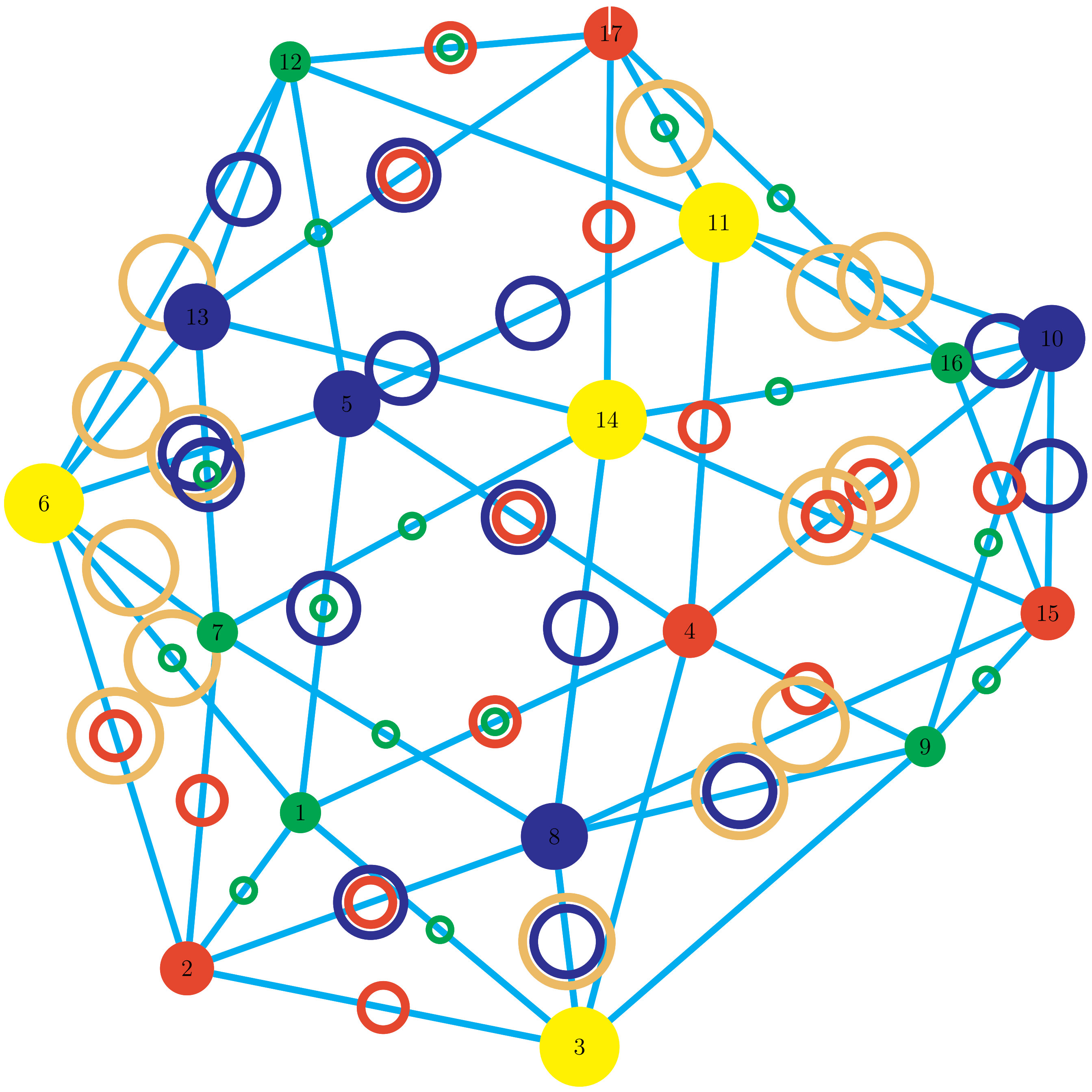

Figure 131. Instructions: 483: color edge 12–5 Green,

Finally we have properly four colored the graph.

2. Observations

It is property B that makes it easy to find the next alternating chain because it eliminates the problem of self conflicting chains. At certain points it looks as though you could grab extra territory by placing an extra circle on an edge without violating property A, or B. But this would be a mistake. By adding an extra circle gratuitously you would have spent a cartridge with nothing to show for it in the form of work done, a gross inefficiency. The algorithm is designed to break the problem into components and to solve one component at a time. Since it is easy to find the right alternating chain by sight for a human, it should be "easy" algorithmicaly.

3. Future Work

This example was made by writing individual instructions which were then executed by a computer program. This needs to be automated piecemeal so that the the algorithm handles everything without the need for human intervention. It will take a lot of work to arrive at a fully fleshed out algorithm, a lot of programming and testing, too much work for a single individual I think. I believe that if people give it a try they will like how nicely it works. This would be a great project for students, a lot of progress can be made just by using the method of individual instructions used here. I would be very interested in any counterexamples. The ultimate goal would be a completed algorithm, which I believe would provide a simple proof of the four color theorem.

email: johnclairmont727@gmail.com