Log N Algorithm for Search from Unstructured List

Dhananjay P. Mehendale
Sir Parashurambhau College, Tilak Road, Pune 411030,
India
dhananjay.p.mehendale@gmail.com

Abstract

The unstructured search problem asks for search of some predefined number, called target, from given unstructured list of numbers. In this paper we propose a novel classical algorithm with complexity $\sim O(LogN)$ for searching the target from unstructured list of numbers. We thus propose a new algorithm, which achieves improvement of exponential order over existing algorithms. Suppose N is the largest number in the list then we consider N dimensional vector space with Euclidean basis. With each of the numbers in the given unstructured list we associate the unique basis vector among the vectors that form together the Euclidean basis. For example suppose j is a number in the list then we associate with this number j the unique basis vector in the above mentioned N-dimensional vector space, namely, $|j>$ = transpose(0, 0, 0, … , 0, 0, 1, 0, 0, … , 0, 0, 0), where there is entry 1 only at j-th place and every where else there is entry 0. We then divide the given list of numbers in two roughly equal parts (i.e. we divide the given bag, B say, containing scrambled numbers in two roughly equal parts and put them in two separate bags, bag B1 and bag B2. We represent the list of numbers in bag B1, bag B2 in the form of equally weighted superposition of basis vectors associated with the numbers contained in these bags, namely, we represent list in bag B1 (bag B2) as a single state formed by equally weighted superposition of orthonormal basis states forming Euclidean basis corresponding to numbers in the bag B1 (bag B2), namely, $|\psi_1>$ ($|\psi_2>$). Let t be the target number. It will be represented as basis state $|t>$ called target state. We then find the value of scalar product of target state $|t>$ with $|\psi_1>$ (or $|\psi_2>$). It will revel us whether t belongs to bag B1 (or bag B2) which essentially enables us to carry out the binary search and to achieve above mentioned $\sim O(LogN)$ complexity!

1. Introduction: The unstructured search problem requires us to find a particular target item amongst a set of N candidates. We can label these N candidates by indices x in the range $1 \leq x \leq N$, and we are supposed to find the index of the sought after target item, $x = t$, say. Now, suppose these N numbers, as tags associated with items as identifiers for these items, be mixed randomly among each other in a bag. Your task is to pick out the target, $x = t$, in fewest possible trials from this bag containing this randomly done mixture of numbers. It is this random mixing of numbers which makes this list of numbers unstructured. This is essentially the well-known so called problem of unstructured search. Also, it is well known that the existing classical algorithm for the solution of this problem has complexity $\sim O(N)$ and Grover's quantum algorithm developed by Lov Grover [1] has complexity $\sim O(\sqrt{N})$.

In this paper we propose a novel classical algorithm which achieves exponential speedup over existing algorithms. The so called binary search algorithm can determine the number which is declared as target from the given list of N numbers with complexity of the order of $\sim O(LogN)$ but this algorithm works only when given list of numbers is sorted (ordered).

For our new algorithm we begin with associating a unique unit vector from the standard Euclidean basis with each number in the given scrambled bag of numbers, thus, we consider each number in the bag as a unique basis vector in the N-dimensional Euclidean space when the largest number in the given scrambled list of numbers in the bag is N. Thus, each number j in the given list is represented as state $| j >$, a column vector, where,

$$| j >= \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

It is important to note that all the scalar components of this vector are 0 except scalar component in the place j from top which is equal to 1. Further, we associate equally weighted superposition of basis vectors with given list of numbers. Thus, we represent entire list as a single superposed state. Let $i_1, i_2, i_3, \cdots, i_R, \cdots, i_N$ be the numbers in the bag and there is no order relation among these N numbers, then we represent this entire list as a single state, $| \psi >$, obtained as superposition state:

$$|\psi>=|i_1>+|i_2>+|i_3>+\cdots+|i_R>+\cdots+|i_N>$$

We further consider projection of state $|\psi>$ on the target state, $|t>$. In other words, we consider scalar product $<t|\psi>$. This scalar product will obviously satisfy the following conditions:

$<t|\psi> = 0$, if state $|t>$ doesn't belong to basis states forming state $|\psi>$

and

$<t|\psi> = 1$, if state $|t>$ belongs to basis states forming state $|\psi>$

This simple fact will enable us to carry out the desired binary search on the scrambled list of numbers. We will see how in the algorithm given below.

2. Algorithm:

1. Let B be the bag containing scrambled list of numbers $\{i_1, i_2, i_3, \cdots, i_R, \cdots, i_N\}$ and let N be the largest among the numbers in this given bag B, and let t be the number which is target to be searched.
2. Consider N-dimensional Euclidean vector space, E, with Euclidean basis of unit vectors. Thus, a basis state $|j> =$ transpose (0, 0, 0, …, 0, 1, 0, …, 0), where all components are 0 except j-th component which is 1. Note that with each number j in the bag we will be associating state $|j>$ in the Euclidean basis.
3. Divide list in bag B, in any arbitrary way, into two sub-lists of roughly equal sizes N_1 and N_2, and put them in bags B1 and B2. Let now bag B1 contain scrambled numbers $\{i_{11}, i_{12}, \cdots, i_{1p}\}$ and bag B2 contains scrambled numbers $\{i_{21}, i_{22}, \cdots, i_{2q}\}$, where, $p \approx q$ from the original scrambled list.
4. Form any one of the following superposed states:
$$|\psi_1>=|i_{11}>+|i_{12}>+\cdots+|i_{1p}>,$$
$$(\text{or, } |\psi_2>=|i_{21}>+|i_{22}>+\cdots+|i_{2q}>)$$
and consider scalar product $<t|\psi_1>$ (or $<t|\psi_2>$)
5. Suppose we have formed state $|\psi_1>$ and if $<t|\psi_1> = 1$ then state $|t>$ is member of the superposition of states that forms state $|\psi_1>$. Else, if $<t|\psi_1> = 0$ then state $|t>$ is member of the superposition of states that forms state $|\psi_2>$.
6. When $<t|\psi_1> = 1$, then set B1 = B and go to step 3. When $<t|\psi_1> = 0$, then set B2 = B and go to step 3.

7. Continue till We (obviously) reach the desired target state and thus the number which is target.

3. Example: Suppose we are given {2, 11, 7, 5, 3, 6, 9, 4} as scrambled list of numbers in Bag B and suppose number 3 is our target, i.e. we wish to locate and find number 3 in this scrambled list. As per steps of algorithm we divide these numbers, in bag B say, into two bags B1, B2 such that bag B1 contains numbers {2, 11, 7, 5} and bag B2 contains numbers {3, 6, 9, 4}. Since 11 is the largest number in the list of given numbers we consider 11-dimensional Euclidean vector space and form state $|\psi_1\rangle = |2\rangle + |11\rangle + |7\rangle + |5\rangle$
Further, we find scalar product $\langle 3|\psi_1\rangle$, which is equal to 0. So, clearly, target 3 belongs to bag B2. So, we set B2 = B and proceed with step 3, i.e. the division of this newly defined bag B. In other words, we will then have newly defined bags B1 = {3, 6} and B2 = {9, 4}. We form state $|\psi_1\rangle = |3\rangle + |6\rangle$ and again find scalar product $\langle 3|\psi_1\rangle$ which is equal to 1. So, clearly, target 3 belongs to bag B1. So, we set B1 = B and proceed with division of this newly defined bag B. In other words, we will have now newly defined bags B1 = {3} and B2 = {6}. We form state $|\psi_1\rangle = |3\rangle$ and again find scalar product $\langle 3|\psi_1\rangle$ which turns out to be equal to 1. So, we have thus obtained the desired target!

Acknowledgements

Thanks are due to Prof. M. R. Modak and Prof. P. S. Joag for useful discussions.

References

1. L. K. Grover, A Fast Quantum Mechanical Algorithm for Database Search, ACM Symposium on the Theory of Computing, ACM Press, New York, pp. 212-219, 1996.