

Yuan Gao z5239220 Q2

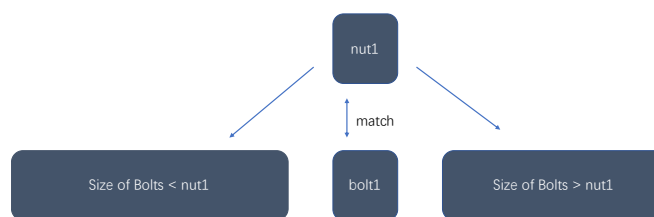
Suppose that you bought a bag of n bolts with nuts screwed on them. Your 5 years old nephew unscrewed all the nuts from the bolts and put both the nuts and the bolts back into the bag. The bolts are all of similar quite large size but are actually of many different diameters, differing only by at most a few millimeters, so the only way to see if a nut fits a bolt is to try to screw it on and determine if the nut is too small, if it fits or if it is too large. Design an algorithm for matching each bolt with a nut of a fitting size which runs in the expected time $O(n \log n)$. (20 points)

We can build an abstract model:

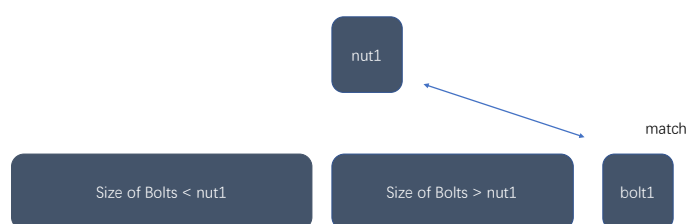
Suppose there are two arrays $\text{nuts}[n]$ and $\text{bolts}[n]$ which have same size elements but out of order.

Step1: pick one nut from $\text{nuts}[n]$

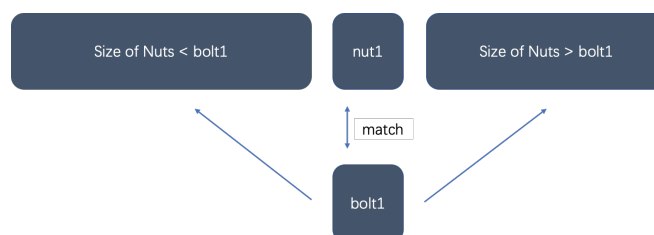
Step2: use this nut to compare with bolts.



Step3: move bolts to right



Step4: use this bolt (has matched) to compare with nuts:



Step5: move nut to right



Step6: repeat step1-5(recursion)

Use a dichotomy similar to quick sort

Therefore, the expected time is $O(n \log n)$

```
1. //This is the cpp code
2. /*Swap two value*/
3. void swap(int * a, int * b){
4.     int tmp;
5.     *a = tmp;
6.     *a = *b;
7.     *b = tmp;
8. }
9.
10. void machBN(int *bolts, int *nuts, int left, int right){
11.     if (left < right){
12.         /*Step1: pick one nut from nuts[n]*/
13.         int tmp = nuts[right];
14.         int i = left;
15.         int j = right;
16.         /*Step2: use this nut to compare with bolts. */
17.         while(i < j){
18.             while(i < j && bolts[i] < tmp)
19.                 i += 1;
20.             while(i < j && bolts[j] > tmp)
21.                 j -= 1;
22.             if(i < j){
23.                 swap(bolts[i], bolts[j]);
24.             }
25.         }
26.     }
27.     /*Step3: move bolts to right*/
28.     swap(bolts[right], bolts[i]);
29.
30.     i = left;
31.     j = right;
```

```
32.      /*Step4: use this bolt (has matched) to compare with nuts:*/
33.      tmp = bolts[right];
34.      while(i < j){
35.          while(i < j && nuts[i] < tmp)
36.              i += 1;
37.          while(i < j && nuts[j] > tmp)
38.              j -= 1;
39.          if(i < j){
40.              swap(nuts[i], nuts[j]);
41.          }
42.      }
43.      /*Step5: move nut to right*/
44.      swap(nuts[right], nuts[i]);
45.      /*Step6: repeat step1-5(recursion)*/
46.      machBN(bolts, nuts, i, right - 1);
47.      machBN(bolts, nuts, left, i - 1);
48.  }
49. }
```