Xuan Gong
xgwt2@mst.edu
573-308-9612

ANNOUNCEMEDIA
Delivering Consumer Insights

Algorithm Challenge –

The rules are simple, complete this challenge by end of day and return your answers to our table to be eligible to win an IPod Touch. If you can't get this done before we leave, complete and email to recruiter@announcemedia.com to be eligible to win the IPod Shuffle. We know what answers are on Google, so be sure to do this yourself.

1. Given a prime number, N, write an algorithm to get the next prime number.

Prime number is odd number. for given prime number N, to find the next prime number, we can search it from N+2 to 2N-1. with step 2. In this case, we can get rid of all the even number.. We assume Next prime number is Q

```
step 1. check ⌊sqr N⌋
  if it ≤ 2
  use brute-force method.
    N=1, Q=2
    N=2, Q=3
    N=3, Q=5
    N=5, Q=7
    N=7, Q=11

step 2. if the value > 2.
    Q = N+2i  (Q should be ~~from~~ N+2 to 2N-1)
    we need to find value i
    initial i=1.
    if(check prime (Q)= true)
        return Q
    otherwise i++

check prime (Q)
1. t= ⌊sqr(Q)⌋
for(i=2, i≤t-1, i++)
{ if (t%i==0)
        return false
}
return true
```

2. Given an array of red, black, and yellow balls, sketch an algorithm to produce 3 arrays of balls, each containing all balls of a single color.

1. We split the array into three parts with same size.

2. In this case, we pick the first ball of each parts, and we can check them at the same time.

3. procedure to add related balls:

    If three balls are different, we can add balls to related arrays at one action.

    If two same color balls, and one other balls, we can add to arrays at two actions

    If three same color balls, we can add at three actions

until finish all balls.

Best case: $O(\frac{N}{3})$.

Worst case. $O(3N)$

3. Given a file of 250 million strings, with a minimum of 200 characters per string, sketch an algorithm to find a string that appears at least twice.

key problem here is the size of file (250 million) $\rightarrow N$ is far larger than size of string (200) $\rightarrow n$.

So, when I compare the strings, I can use bruteforce algorithm ($O(n^2)$)

If I use skip search algorithm, the time complexity can reduced to $O(n)$.

But this is not the critical problem here, the critical problem is how to find a string appears at least ~~twice~~ in this big file (250 million strings)

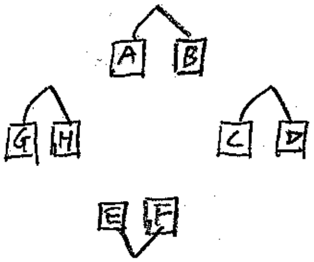
(Algorithm will be given on the back)

Recruiting@announcemedia.com

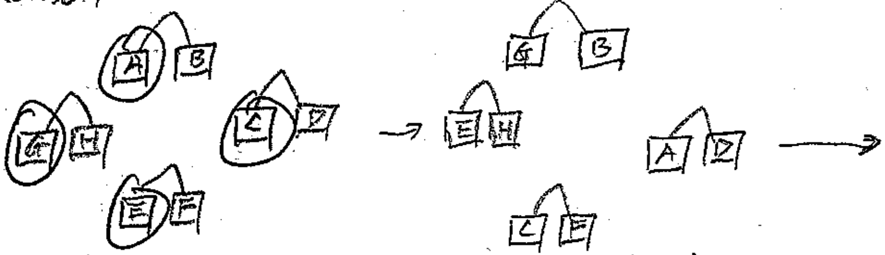
www.announcemedia.com

① we split N strings into $\frac{N}{2}$ groups, each groups has 2 string, and organize as a cycle.

② Procedure of comparing:

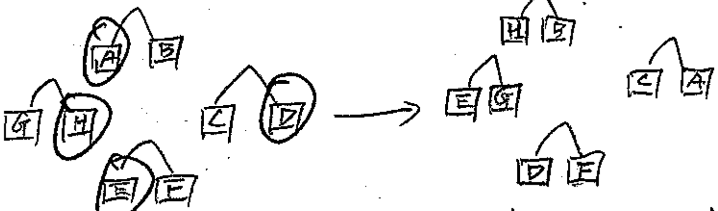


I the first run. pick the first string in each group, compare and rotate in clockwise, such as



after $\frac{N}{2}$ movement, it will back to original place, so, each string has compare $\frac{N}{2}$ string. in the process. If we find any same string, we can return

II The second run, pick the first string in first group, second string in the second group, and first string in third group, second string in fourth group, so on and so forth.



after $\frac{N}{2}$ movement, it will back to original place, so each string has compare another $N/4$ strings.

We can find each groups of string has do the comparion with their neighborhood groups, of strings. such as (A B) has compared with (C D) (G H), but not (E F).

III Next, we even split these groups into anther two big groups and do the same process into these two groups at the same time

$G_1 = \{$ group 1, group 3, group 5, - - - - $\}$, $G_2 = \{$ group 2, group 4, group 6. - - - $\}$

Do step I, II, III in G_1, G_2 simultaneously, continue until each big group G has only two small groups, we compared every strings with other strings