Algorithm Design and Analysis: Homework 1

Due on October 10, 2017 at 9:00am

14784547 luochenqi

Luochenqi

Problem 1

Solution

$g_1(n) = 2^{\sqrt{logn}} = o(n^{\frac{4}{3}}) = o(n(logn)^3) = o(2^n) = o(2^{n^2}) = o(2^{2^n}) = o(n^{logn})$

Problem 2

Solution

(a)O(f(n))=n

(b)we can use following method to decrease the time complexity.

```
int exp(int x,int n){
    if (n==1) return x;
    if (n%2==0){
        return exp(x,n/2)*exp(x,n/2);
    }else{
        return exp(x,(n-1)/2)*exp(x,(n-1)/2)*x;
    }
}
```

so O(g(n))=log N so $\lim_{x\to+\infty}\frac{g(n)}{f(n)} = 0$

Problem 3

Solution

L:Empty list that will contain the sorted elements

S:save the all node in the S set with no incoming edgess

```
while S is not empty
{
    remove a node n from S
    add n to tail of L
    for each node m with an edge e from n to m do
        if length(S)>=2 return it does not exist a path
        remove edge e from the graph
        if m has no other incoming edges then
            insert m into S

}
if graph has edge then return it has at least one cycle
    else
        return L
```

Problem 4

Solution

The question means if we delele the spot it still will be a strongly connected graph.
For the root node if has two or more subtrees it can't be a spot for toilet.
but if it is not the root node ,we can use two array to compare to check if it can be the spot.
we define dfn[i] to monitor the node's dfs order
and low[i] to take the first node we can recall in order.

so $low[u] = \begin{cases} min(low[u], low[v]) & \text{(u,v) is a tree edge} \\ min(low[u], dfn[v]) & \text{(u,v) is a recall edge and v is not u 's father} \end{cases}$

so when low[v] $\geq$ dfn[u] it is a spot that can not be toilet.

```
for (u,v) in E
    if v has not found
        children++;
        dfs(v)
        low[u]=min(low[u],low[v])
        if parent[u]=null &&children>=2
            then u spot can not be the toilet
        if parent[u]!=null && low[v]>=dfn[u]
            then u spot can not be the toilet
    else if (v is not root node)
        low[u]=min(low[u],dfn[v])
```

Problem 5

Solution

the level between s and t must has a single node level.because if it is has not one single node level. $\frac{n-2}{2}+1=\frac{n}{2}$it could not larger than $\frac{n}{2}$,so it must has a single node which is between s and t ,so if we delete this node ,it will destroy all s-t paths. we can use BFS to find the node that we have visited frequently.then we can find the node

Problem 6

Solution

We can sort the expectation from small to big and sort the actual cake from small to big and give the first actual cake to the first children and do in the same manner. because if you give the cake to the second man and he is satisfied the first must also be satisfed,so it is better for you to give the first man to add one satisfaction.

Problem 7

Solution

Firstly,we sort the list of points such that $x_1 \leq \cdots \leq x_n$,if the list is not empty select the leftmost point in the x1 to made the interval$[x_1, x_1 + 1]$ let the $x_1, x_2 ... x_1 + k$ out from the list.

None of the intervals obtained by smallest-set can move right by any distance, otherwise some points wont be covered.

proof:Suppose none of the first m intervals can move right. When A[j]¿B[m]+1 last interval [B[m],B[m]+1]cannot cover A[j].So we need a new interval let it be [A[j],A[j]+1]which cannot move right ,or A[j]cannot be covered.