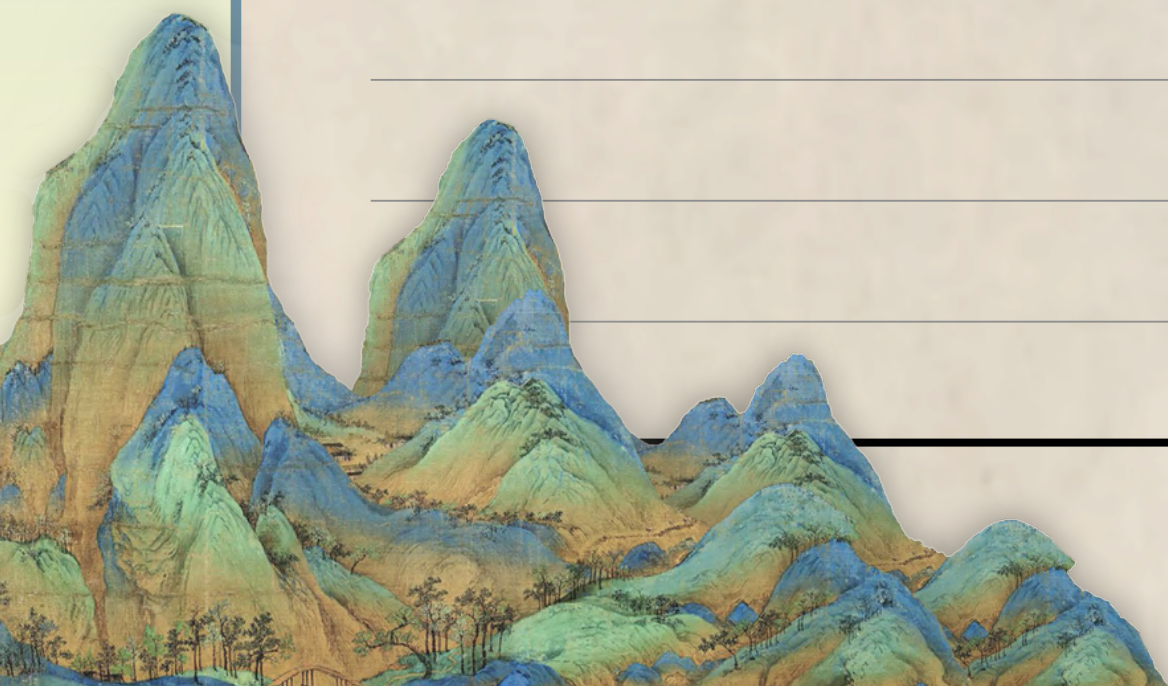


主題	日期
	星期

Homework 3 3.2, 3.8, 4.4, 4.6, 4.8, 4.18
謝澤鈞 2020012544

3.2 We need a little improvement of BFS:

```
Set Discovered[s] = true and Discovered[v] = false for all other v
Set From[v] = null for all nodes v
Initialize L[0] to consist of the single element s
Set the layer counter i = 0
Set the current BFS tree T = ∅
While L[i] is not empty
    Initialize an empty list L[i+1]
    For each node u ∈ L[i]
        Consider each edge (u,v) incident to u
        If v = s then
            Set the temporary variable temp = s
            While From[temp] is not s then
                print(temp)
                temp = From[temp]
            Elseif Discovered[v] = false then
                Set Discovered[v] = true
                Add edge (u,v) to the tree T
                Add v to the list L[i+1]
            Endif
        Endfor
    Increment the layer counter i by one
Endwhile
```



主題

日期

星期

3.8 *First let's prove two lemmas:

Lemma 1: $\text{dist}(u,v) + \text{dist}(v,w) \geq \text{dist}(u,w)$

(u, v, w are independent nodes in G)

Lemma 2: $\text{dist}(u,v) \geq \text{dist}(u,w) - \text{dist}(v,w)$

These two lemmas are easy to prove.

* Now let's back to the original problem:

Suppose a Graph $G = (V, E)$ such that $|V| = n$, $\text{diam}(G) = d$

Then we have $x, y \in V$, $x \neq y$, $\text{dist}(x, y) = d$

For any $z \in V$, $z \neq x, y$, by lemma 1 we have:

$$\text{dist}(z, x) + \text{dist}(z, y) \geq \text{dist}(x, y) = d$$

$$\Rightarrow \max\{\text{dist}(z, x), \text{dist}(z, y)\} \geq \frac{d}{2} \quad ①$$

Thus, we can find $\{z_1, z_2, \dots, z_k\} \subseteq V$ ($k = \lfloor \frac{d}{2} \rfloor$)

such that $\text{dist}(z, z_i) = i$ ($1 \leq i \leq k$)

$$\text{Sum them up we have } \sum_{i=1}^k \text{dist}(z, z_i) = \frac{k(k+1)}{2} \geq \frac{d^2-1}{8}$$

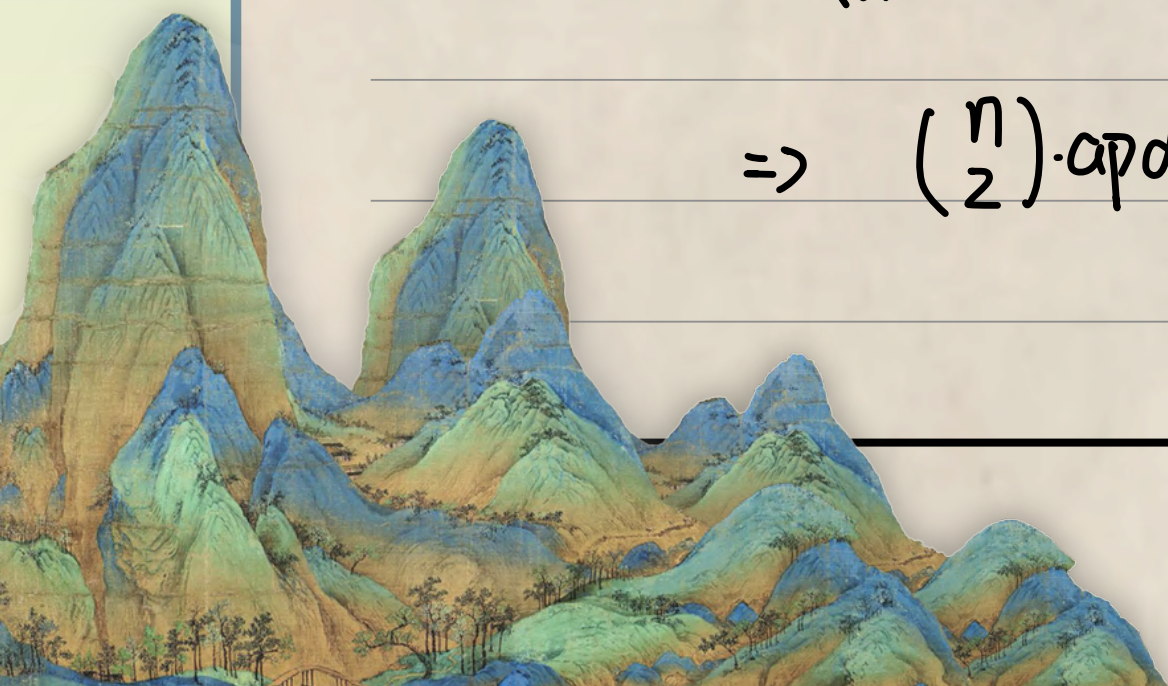
$$\text{Generally, } \sum_{w \in V} \text{dist}(z, w) \geq \frac{d^2-1}{8}$$

Since we choose z randomly, we have

$$\sum_{z \in V} \sum_{\substack{w \in V \\ w \neq z}} \text{dist}(z, w) \geq \frac{n(d^2-1)}{8}$$

$$\Rightarrow \sum_{(u,v) \in E} \text{dist}(u,v) \geq \frac{n(d^2-1)}{16}$$

$$\Rightarrow \binom{n}{2} \cdot \text{apd}(G) \geq \frac{n(d^2-1)}{16}$$



主題

日期

星期

$$\Rightarrow \text{apd}(G) \geq \frac{d^2-1}{8(n-1)}$$

$$\Rightarrow \frac{\text{diam}(G)}{\text{apd}(G)} \leq \frac{8d(n-1)}{(d^2-1)} \leq \frac{8n}{d}$$

Emm... Sorry but I cannot estimate it more precisely :)

4.4 Suppose a string s' and a string s , whose length is relatively m and n ($m \leq n$)

Now we introduce the following algorithm

First, suppose an array $f = (f_1, f_2, \dots, f_m)$ generated by the following algorithm:

let $i=1, j=2$

let $f_1=1$

while $i < j \leq m$:

if $s'_j = s'_i$ then:

$i=i+1; j=j+1; f_j = f_{j-1}+1$

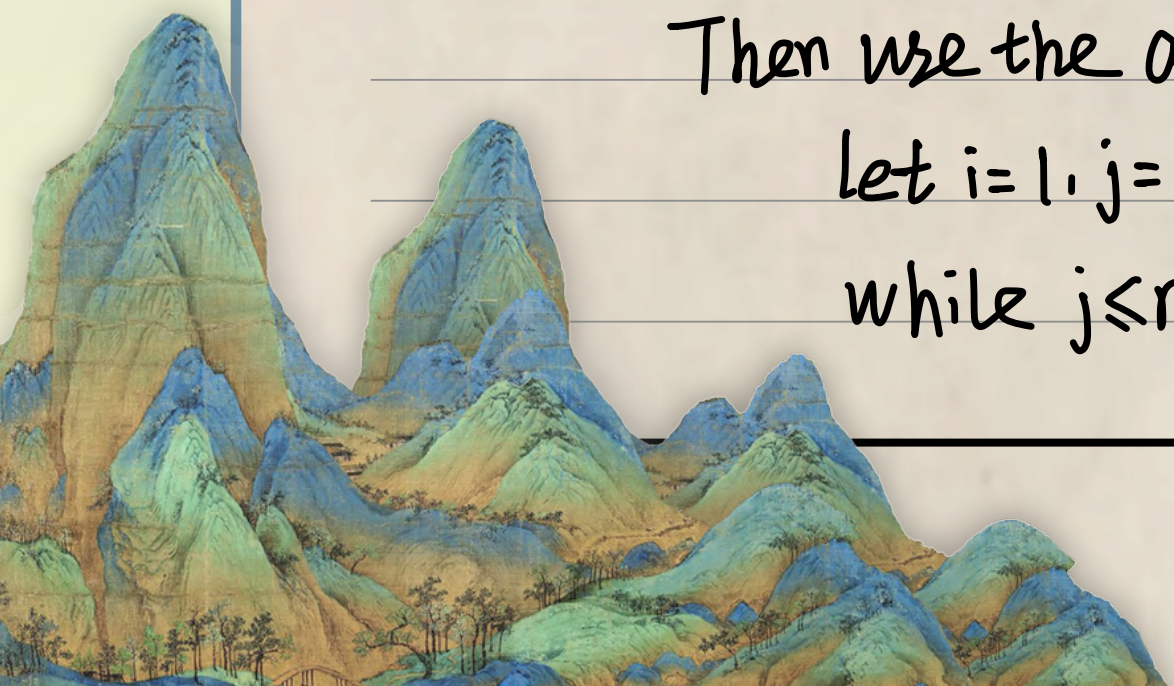
else $i=f_{i-1}$

According to the algorithm we know that f_i means the position where we can examine if we find s'_i doesn't fit s at the position;

Then use the array f we can continue the following process:

let $i=1, j=1$

while $j \leq m$



主題

日期

星期

```

if  $S_i = S_j$  then  $i = i+1, j = j+1$ 
else  $i = i, j = j-1$ 
end while

```

```

if  $j = m+1$  then return "true"
else return "false"

```

And the time complexity of this algorithm is $O(m+n)$

4.b. Our algorithm is simple.

For each contestant i , we suppose his/her swimming time is t_i , and his running and cycling time is t_i'

Then we draw the conclusion that:

The contestant starts at first ends at $t_1 + t_1'$

The contestant starts then ends at $t_1 + t_2 + t_2'$

...

The contestant starts in the end ends at

$t_1 + t_2 + \dots + t_n + t_n'$

So our goal is to find a best order $1, 2, \dots, n$

such that $\max_{1 \leq i \leq n} (t_1 + t_2 + \dots + t_i + t_i')$ reaches the

lowest value.

Following is an important conclusion:

For any $1 \leq i < i+1 \leq n$, there exists a best order where

$$t_i - t_i' \leq t_{i+1} - t_{i+1}'$$

主題

日期

星期

In fact, the No. i and No. $(i+1)$ starter separately ends at $(t_1 + \dots + t_{i-1}) + t_i + t_i'$ and $(t_1 + t_2 + \dots + t_{i-1}) + t_i + t_{i+1} + t_{i+1}'$

the later ending time of these two is $\max \{ (t_1 + t_2 + \dots + t_{i-1}) + t_i + t_i', (t_1 + \dots + t_{i-1}) + t_i + t_{i+1} + t_{i+1}' \}$

However, if we swap them two then it becomes $\max \{ (t_1 + t_2 + \dots + t_{i-1}) + t_{i+1} + t_{i+1}', (t_1 + \dots + t_{i-1}) + t_{i+1} + t_i + t_i' \}$

We want the first one to be lower, then we need:

$$t_i + t_{i+1} + t_{i+1}' \leq t_{i+1} + t_i + t_i'$$

$$\Rightarrow t_i - t_i' \leq t_{i+1} - t_{i+1}'$$

So according to this conclusion, we have the following:

Sort all the candidates with the value $t_i - t_i'$ from the lowest to the biggest.

This is one of our needed orders.

4.8. We have proved that, if G is divided into two parts S and S' , then the shortest edge connecting S and S' is included.

Suppose e_1 is the shortest edge among all. then $e_1 \in T$ (T is any of the spanning trees)

Delete one of the two nodes e_1 connects, remaining a new graph G' with $(n-1)$ nodes.

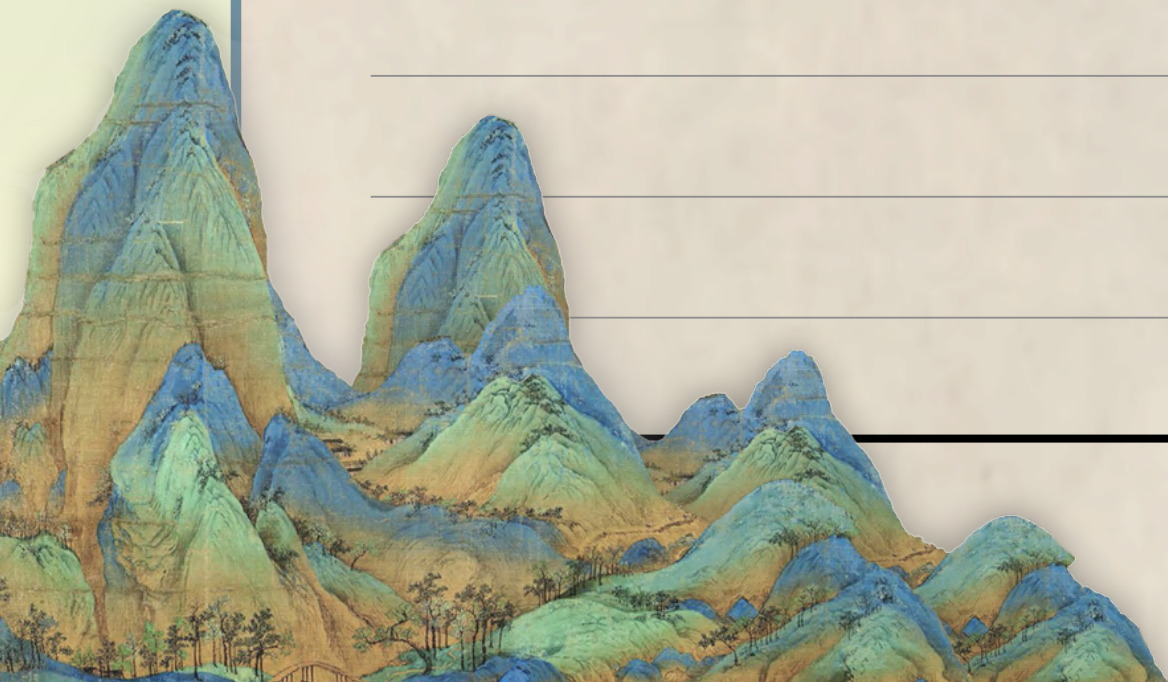
主題	日期
	星期

Find the shortest edge in G_i and delete one of the two nodes it connects.

...

Through $n-1$ steps like this we can finally get the spanning tree T .

Since each of the $(n-1)$ nodes in T should be included by all spanning trees, we have T the unique spanning tree.



主題	日期
	星期

