

VE 477 Homework 3

Wang Yichao, ID: 517370910011

- **Exercise 1.** 1. Similar to BFS, DFS is used for search problems. DFS has the priority of depth and requires less space complexity compared to BFS.
2. Topology sorting is used for directed acyclic graph(DAG). if $A \rightarrow B$, then A is sorted before B.
3. Hamiltonian path is NP.

Algorithm 1 Hamilton

Input: a graph $G(V, E)$ which is DAG

Output: whether contains a Hamilton path

▷ first topology sort then judge

```
1: function TOPOLOGY( $G(V, E)$ )
2:    $L = \{\}$ 
3:    $S =$  nodes without incoming edge
4:   while  $S$  is not empty do
5:     choose a node  $n$  from  $S$ 
6:     remove  $n$  from  $S$ 
7:      $L = L \cup \{n\}$ 
8:     for node  $m$  with edge  $e = n \rightarrow m$  do
9:       remove  $e$  from  $E$ 
10:      if  $m$  has no incoming edge then
11:         $S = S \cup \{m\}$ 
12:      end if
13:    end for
14:  end while
15:  return  $L$ 
16: end function
17:  $L = \text{topology}(G(V, E))$ 
18: for  $i = 0 : |L| : 1$  do
19:   if  $L[i] \rightarrow L[i + 1] \in E$  then
20:     Continue
21:   else
22:     return False
23:   end if
24: end for
   return True
```

4. it is obvious that line 17 to line 23 has complexity of $\mathcal{O}(|V|)$.

for topology sort, we notice that the while loop has $\mathcal{O}(|V| + |E|)$, the remove of edge means that we has $\mathcal{O}(|E|)$. We should be careful about the insert m to S step, otherwise the while loop has only $\mathcal{O}(|V|)$. In all, the time complexity is $\mathcal{O}(|V| + |E|)$.

5. The decision problem in this question is only in P. But the Hamilton path problem is NP-complete.

- **Exercise 2.**

1. no, using stirling's formula, $n! \sim n^n e^{-n}$, thus $\log n! \sim \log n^{\log n} e^{-\log n}$. However, $\log n^{\log n} \gg n^c$, where c is a constant.

2. another hard problem directly copied from CLRS TAT, maybe this is the reason that the homework will not change until year CLRS.

The former one is asymptotically larger. assume $\log^* n = k$, then we have $\log^* \log n = k - 1$ and $\log \log^* n = \log k$. The rest comparison is trivial.

3. 3 tests are enough.

Algorithm 2 weight test

Input: 8 balls b1 to b8, with one much lighter

Output: the lighter ball

- 1: assume weight(x) return the weight of x
 - 2: compare weight(b1 b2 b3 b4) and weight(b5 b6 b7 b8)
 - 3: assume the lighter set is c1 c2 c3 c4
 - 4: compare weight(c1 c2) and weight(c3 c4)
 - 5: assume the lighter set is d1 d2
 - 6: compare weight(d1) and weight(d2)
 - 7: assume the lighter set is d1
 - 8: **return** d1
-

• **Exercise 3.** reference:

1. https://en.wikipedia.org/wiki/Rubik%27s_Cube
2. https://en.wikipedia.org/wiki/CFOP_method
3. https://en.wikipedia.org/wiki/Lars_Petrus

The game:

the rubik's cube is a game. in this game, we have a cubic which has 6 surfaces. on each surface, we divide the square into 9 small blocks. so we have 6 times 9, which is 54 blocks evenly distributed in 6 surfaces. then we color each surface with a unique color. Figure 1 shows a ordered cubic. Then we rotate the cubic along axis to break its order. Figure 2 shows a misordered cubic. The aim of this game is usually to recover the misordered cubic within the shortest time.



Figure 1: order



Figure 2: disorder

Solving algorithm 1.

CFOP: we need four steps, the cross, first two layers, orientation of the last layer, permutation of the last layer. first, we set a base surface and form a cross on it. second, we notice that the cross divide the first two layers below into four areas. third, we deal with the last layer and makes the surface opposed to our base in the same color. Last, we deal with the third layer and recover it.

Solving algorithm 2.

Roux Method: 6 steps in total. 1. we build a 2x2x2 block. 2. we extend this block into 2x2x3, but we should not destroy the original 2x2x2 part. 3. correct edge orientation. 4. complete two layers. 5. permute and orient so that the remaining corners are solved. 6. permute the last edge

- **Exercise 4.** 1. if we can prove that there does not exist a polynomial algorithm that solve the simple path, we can be mathematicians. i will only focus on the certificate. Luckily, in this question we only need to prove NP. Let the certificate to be the path P. we can simply use $\mathcal{O}(|V|)$ to check if the edges on path is in the graph and if the vertices are repeated. This is polynomial and in P. Thus the problem is in NP. Qed.

2. the certificate would be the factor. with the certificate, we only need to find the gcd of the number and the factor. as we know from ve 475, gcd is done in polynomial time, thus NP. Qed. without this hint, this is equivalent to deciding if it is a prime. Qed.

3. the certificate would be the k specific vertices. For each of them, remove the edges that contains the vertex. After k rounds, we check if the edges are removed clearly. if not, the solution is wrong. this only take time complexity of $\mathcal{O}(k \cdot |E|)$ or $\mathcal{O}(|V|^2)$, since there are at most $\mathcal{O}(|V|^2)$ edges in the graph. this polynomial time complexity means NP. Qed.

- **Exercise 5.** No.

Prime Number Theorem says that the probability of a random integer being smaller than N being prime is close to $1/\log n$.

Assume we have an n digit number, then under decimal. $\sqrt{10^n}$ is not a polynomial time for size n , it is exponential time. However, we can use smarter polynomial algorithms to cope with PRIME problem, see papers for more detail if you want to be a mathematician.