CS6033 Assign No.4

Minghe Yang

March 2021

1 Time vs. space

1.1 on NUDT Tianhe-2 calculating 2^{64}:

$$\frac{2^{64}}{3386 * 10^{15}} = 544.79s \tag{1}$$

on NUDT Tianhe-2 calculating 2^{80}:

$$\frac{2^{80}}{3386 * 10^{15}} = 35,703,357.44s \tag{2}$$

1.2 Normal cpmputers needed for 2^{64} in one day:

$$\frac{2^{64}}{3.8 * 10^{9} * 24 * 60 * 60} = 56185.26 \tag{3}$$

So 56186 laptop computers are needed.
Normal cpmputers needed for 2^{80} in one month(30 days):

$$\frac{56185.26 * 2^{16}}{3.8 * 10^{9} * 24 * 60 * 60 * 30} = 1.22 * 10^{8} \tag{4}$$

1.3 Hard drives needed for 2^{64}:

$$\frac{2^{64}}{16 * 2^{40}} = 1048576 \tag{5}$$

Hard drives needed for 2^{80}:

$$1048576 * 2^{16} = 6.9 * 10^{10} \tag{6}$$

2 Critical thinking

2.
for the first few $i \leq k$, implement n_i to the list.
for $i>k$, roll a number d from 0 to i, if$d \in (1, k)$, replace the d_{th} element with n_i.
For the first set, the chance to keep in the list is $k/(k+1) * (k+1)/(k+2) * ... * (n-1)/n = k/n$
For the remaining set, the chance to get in the list will be $k/i * i/n = k/n$.

3 Algorithm and complexity

3.1

Algorithm 1 Layer sum detection

```
Input: number of the layer i
Output: the sum of all inside units
return $3^{i-1}$
```

3.2 the complexity is $O(1)$ since it's a constant.

4 Clique problem

5.2 Given a graph and a clique of k vertices, for the worst situation it will cost you $O(n^2)$ to solve the question, with a right hint, you can solve it in polynomial time, so it is NP problem.

5.3
What we need to do is to create a k clauses graph to represent this graph. First we create C_{j1}, C_{j2}, C_{j3} for every combination, and draw lines exactly like the original setting.
Then we remove the nodes within every combination since we don't need them.
At last remove edges that join two vetices that are totally negate to others

5.4
It's a NP complete question.

5 IND-SET problem

6.2
An independent set is a set of vertices in a graph, no two of which are adjacent. It is a set of vertices such that for every two vertices in the graph, there is no edge connecting the two.

6.3
If you want to check it, the hardest solution should be $O(n^2)$ (check between every edge) while you can check it's correctness in polynomial time.

6.4
What we need to do with the original graph is
1.Draw all possible edges that are not in the original graph with a different color, so then every vertex is linked to others.
2.Delete all the original edges.

6.5 It is a NP-hard question.