VE 477 Homework 4

Wang Yichao, ID: 517370910011

- **Exercise 1.** 1. PFLOPS means 10^15 operations per seconds, so for 2^{64} we need $\frac{2^{64}}{10^{15}\times 33.86} = 544$ seconds, for 2^{80} we need $\frac{2^{80}}{10^{15}\times 33.86} = 3.57\times 10^7$ seconds, which is 413 days.

  2. If the desktop computers can run in parallel, we will need $\frac{2^{64}}{24\times 3600\times 3.8\cdot 10^9} = 56185$ computers. For 2^{80} operations, we would need $\frac{2^{80}}{24\times 3600\times 3.8\cdot 10^9\times 30} = 1.23\times 10^8$ months.

  3. For 2^{64}, we need $\frac{2^{64}}{16\times 2^15} = 3.5\times 10^{13}$ hard drives. For 2^{80}, we need $\frac{2^{80}}{16\times 2^15} = 2.3\times 10^{18}$ hard drives.

  As mentioned in VE475, VE477 and VE472, the space is more expensive compared to time.

- **Exercise 2.**

  random is beautiful. this is a famous example of random algorithm. The basic idea is that we select the first k elements as a[1] to a[k]. Then for every element($m > k$) after, we generate a random number t between 1 and m, if the random number is larger than k, we do nothing; else we replace a[t] by the new element m. And we continue on the next element. The time complexity depends on the random number generate function. For naive random generator, we need $\mathcal{O}(1)$, thus the total time complexity for this is $\mathcal{O}(n)$. The correctness would be that for an element in the first k elements, the probability of not being replaced is $\frac{k}{k+1}\times\frac{k+1}{k+2}\times ...\times\frac{n-1}{n} = \frac{k}{n}$. for the element that are added afterwards, the probability of being selected and remained to the last for the mth element is $\frac{k}{m}\times\frac{m}{m+1}\times ...\times\frac{n-1}{n} = \frac{k}{n}$ as well. Thus they shares the same probability.

- **Exercise 3.**

  1.

Algorithm 1 sum problem

Input: the number i which indicates the ith line

Output: the sum of elements on the ith line ▷ power needs to be rewritten by recursion

```
1: function POWER(i)
2:     if i == 1 then
3:         return 3
4:     else
5:         return 3 × $POWER(i-1)$
6:     end if
7: end function
       return POWER(i)
```

  2. it is obvious that we need n multiplication thus the time complexity is $\mathcal{O}(i)$. Correctness is proved by the definition of this triangle. The key is that all the elements contributes three times in the next layer. Assume the element in the middle to be a_0, the elements on the right are $a_1, a_2, ...$ and the elements on the left are $a_{-1}, a_{-2}, ...$ then a_i will contribute to a_{i-1}, a_i, a_{i+1} in the next layer.

- **Exercise 4.**

  we can follow the slides here. `https://cse.iitkgp.ac.in/~palash/2018AlgoDesignAnalysis/SAT-3SAT.pdf`

  $(x_1\vee x_2\vee z1)\wedge(\neg x_3\vee\neg z_1\vee z_2)\wedge(x_4\vee\neg z_2\vee z_3)\wedge(x_5\vee\neg x_6\vee\neg z_3)\wedge(\neg x_1\vee\neg x_2\vee z4)\wedge(x_3\vee\neg z_4\vee z_5)\wedge(\neg x_4\vee\neg z_5\vee z_6)\wedge(x_5\vee x_6\vee\neg z_6)\wedge(x_1\vee\neg x_2\vee z7)\wedge(\neg x_3\vee\neg z_7\vee z_8)\wedge(x_4\vee\neg z_8\vee z_9)\wedge(x_5\vee\neg x_6\vee\neg z_9)\wedge(x_1\vee\neg x_2\vee z_{10})\wedge(x_1\vee\neg x_2\vee\neg z_{10})$

- **Exercise 5.**

  1. Clique problem is given a graph, decide whether there exists a complete subgraph with k points or more.

  2. the certificate would be a given set of k points, we can use $\mathcal{O}k^2 todecidewhetherthesekpointsformsacompletegraph$, which is polynomial when k is large(close to n), or constant time when k is small. So clique problem is NP.

  3. The trick point is to construct a graph. we construct the graph G to have k clusters with 3 nodes each, each cluster corresponds to a clause in the 3CNF expression. Then we connect the edges between all pairs of

nodes from 2 different clusters, but we should not connect the edge with form $x \to \neg x$. we also skip edges with two end points in the same cluster.

Then if 3SAT, we has at least one point from every cluster, then we can have k points from k clusters that forms a subgraph. In reverse, if k-cluster, the key point is that we can not form a k-cluster with two points from the same sluster(because they are not connected by definition). So we have to choose one points from k clusters each. Thus the reverse is true. Qed.

4. based on the result of question 3, we can conclude that clique problem is NP-complete because 3-SAT is NP-complete.

- **Exercise 6.**

  1. independent set is a subgraph where no edges is connected between any two points within the subgraph. maximum independent set is to find a subgraph that is not a subset of another different independent set.

  2. given an undirected graph and an inteeger k, we should decide whether there exist an independent set of size k.

  3. the certificate is a subset of k points, we should check every pair between these k points, the time complexity should be $\mathcal{O}(k^2)$, when k is large(close to n), it is still polynomial. Thus NP.

  4. this equivalence is trivial. just use a complementation graph. we construct G' with the same vertices as G, for edge e in G, if e is connected, then its corresponding edge e' is not, is e is not, then e' is connected. then k-cluster corresponds to an independent set of size k.

  5. based on the result of question 4, we can conclude that IND-SET probelm is NP-complete because clique problem is NP-complete.