UM-SJTU JOINT INSTITUTE

Introduction to Algorithms
(VE477)

Homework #4

Prof. Manuel

Xinmiao Yu
518021910792

Oct. 23, 2020

Q1.

1. For 2^{64} operations, time is $\dfrac{2^{64}}{33.86 \times 10^{15}} = 544.8[s]$

   For 2^{80} operations, time is $\dfrac{2^{80}}{33.86 \times 10^{15}} = 3.6 \times 10^{7}[s]$

2. As in each day, we have 86400 seconds

   For 2^{64} operations, number is $\dfrac{2^{64}}{3.8 \times 10^{9} \times 86400} = 56186$

   For 2^{80} operations, number is $\dfrac{2^{80}}{3.8 \times 10^{9} \times 31 \times 86400} = 1.19 \times 10^{8}$

3. For 2^{64} bits, number is $\dfrac{2^{64}}{16 \times 8 \times 10^{12}} = 1.4 \times 10^{5}$

   For 2^{80} bits, number is $\dfrac{2^{80}}{16 \times 8 \times 10^{12}} = 9.4 \times 10^{9}$

Q2.

```
Algorithm 1: Algorithm R
Input  : set S of size n
Output: subset S' of size k
for $i \leftarrow 1$ to $k$ do
    $S'[k] \leftarrow S.pop()$
end for
for $i \leftarrow k+1$ to $n$ do
    randomly choose an integer $j$ in range [1, i]
    if $j \leq k$ then
        $S'[j] \leftarrow S.pop()$
    else
        $S.pop()$
    end if
end for
```

Prove the probability of selecting each element is k/n by induction.

a. **Base Case:** the first k elements are included as the first $n = k$ members of the result set S', so the $n - seen$ result set includes each element with $\mathbf{Pr}[k/n = 1]$.

b. Assume the current k elements have already been chosen with $\mathbf{Pr}[k/n]$.

c. For the (n+1) element, it was chosen with $\mathbf{Pr}[k/(n+1)]$. Each element is in the result set S' has $\mathbf{Pr}[1/k]$ to be replaced. Then the probability that an element from the $n - seen$ result set is replaced by the $n+1-seen$ result set is $\mathbf{Pr}[\frac{1}{k}\cdot\frac{k}{n+1} = 1/(n+1)]$. Similarly, an element is not replaced is $\mathbf{Pr}[1-1/(n+1) = n/(n+1)]$.

Then, the $n+1-seen$ result contains (1) either an element in the $n-seen$ result and not be replaced ($\mathbf{Pr}[k/n\cdot\frac{n}{n+1} = k/(n+1)]$), (2) or the element is chosen ($\mathbf{Pr}[k/(n+1)]$).

So the proof is complete.

Q3.

1. The algorithm is

```
Algorithm 2: Algorithm
Input  : i-th line
Output: sum on all elements in the i-th line
return $3^{i-1}$
```

2. Time complexity with input i: $\mathcal{O}(3^i)$.

   Prove the correctness by induction.

   a. **Base:** $i=1$, $3^{i-1}=1$, which is true.

   b. **Assume when** $i=k-1$ **it is true:** then we have $sum(k-1)=3^{k-2}$

   c. **When** $i=k$**:** denote the $n-th$ number in the k row as $N_n(k)$. Then by definition,

   $$N_n(k) = \sum_{x=n-1}^{n+1} N_x(k-1)$$

   Each number in the $k-1$ row will appear three times, so the sum of k row is

   $$\sum_{n=1}^{2k-1} N_n(k) = \sum_{n=1}^{2k-1}\sum_{x=n-1}^{n+1} N_x(k-1) = 3\sum_{x=0}^{2k} N_x(k-1) = 3\cdot 3^{k-2} = 3^{k-1}$$

   Then the result consistent with the assumption, so the correctness is proved.

Q4.

Q5.

1. Given an undirected graph G and positive integer k, determine whether a set S with size k of vertices form a **clique**, such that each vertex in S is adjacent to each other in S.

2. Certificate: a set S consisting of vertices in the clique and S is a subgraph of G.
   Verifier: To check whether exists a clique of size k.

   a. $\mathcal{O}(1)$ to check the item number of S is k.

b. Each vertex should connect to every other vertices, then edge number should be $\binom{k}{2} = \frac{k(k-1)}{2}$. So it takes $\mathcal{O}(k^2) = \mathcal{O}(n^2)$ to check.

Then we could prove the correctness in polynomial time.

3. Assume for each clause $C_1, C_2, ..., C_k$ in F, it has literals x_{j1}, x_{j2}, x_{j3}.

   To construct the graph, for each literal x_{jn}, create a distinct vertex in G. Then let G contains all the edges except

   a. joining two vertices in same clause

   b. joining two vertices whose negation are each other

   To prove F is satisfiable if and only if G has a $k-$clique.
   If F is satisfiable, at least one literal in each clause is True. Pick one such true literal from each clause. Then there will be subgraph constructed by those literals. As each pair of them could not belong to any same clause and could not be negation of each other (because they both are true), there must be edge between them. So this subgraph is a clique of size k.
   If G has a clique with size k, then this clique must have exactly one literal from each clause. Assign True to all the literals corresponding to the vertices in the clique, then F is satisfiable.

4. $\mathcal{NP}-$complete

   Because from 3., 3-SAT could be reduced to Clique in polynomial time.

Q6.

1.

2. Given an undirected graph and a number k, determine whether the graph contains an independent set of size k. A set of vertices inside a graph G is an independent set if there are no edges between any two of these vertices.

3. Certificate: A set of vertices with size k.
   Verifier: Check all vertices in the certificate are in the graph, and no edge between any two of them, run in $\mathcal{O}(V + E)$.

   So the correctness could be verified in polynomial time, it is in $\mathcal{NP}$.

4. Given a graph $G = (V, E)$ has a k-clique, set $G' = (V', E')$ as the complement of G, such that $V' = V$ but each $e \in E \rightarrow e \notin E'$.

   Then from each vertex in $S(clique)$ is adjacent to each other, when exclude all the edges in E, no vertex in S' would have any edge to connect any of other vertices in S'. So "G has a k-clique" is equivalent to "G' has an independent set of size k".

5. $\mathcal{NP}-$complete.

   Because from 3., Clique could be reduced to IND SET in polynomial time, and Clique is $\mathcal{NP}-$complete.