COL351: Analysis and Design of Algorithms

Tutorial Sheet - 6

September 19, 2022

Hashing

Question 1 Let $U = [1, M]$ be a universe, and $S \subsetneq U$ be a fixed set of size n. Present a construction of a random hash function $H : U \to [0, n-1]$ such that

(i) For each $i \in [0, n-1]$, the expected size of $T[i]$ is $O(1)$.

(ii) For each $x \in U$, the time to check membership of x in S is $\Omega(n)$.

Solution: The construction is as follows: (i) Let z be a random number in the range $[0, n-1]$; (ii) Define $H(u) = z$, for each $u \in U$ in the universe.

The expected size of $T[i]$ is $\sum_{s \in S} Prob(H(s) = i) = \sum_{s \in S} \frac{1}{n} = O(1)$.

Further, for any two elements $x, y \in U$, we have $H(x) = H(y)$. This shows that the time to check membership in S is $\Omega(n)$.

Question 2 Let $U = [1, M]$ be a universe, and let $S = \{s_1, \ldots, s_n\}$ be a subset of U of size n such that each s_i is a uniformly random element of U independent of other s_j's. Let H be a hash function such that $H(x) = x \bmod n$.

(i) Show that the expected size of $\left(\max_{i=0}^{n-1} T[i]\right)$ is $O(\log n)$.

(ii) Argue that the expected value of maximum time taken to verify the membership of elements of U in S is $O(\log n)$.

(iii) If we redefine $H(x)$ as $x(\bmod n^2)$, then prove that with probability at least $1/2$, there will be no collisions under H.
Hint: Use Markov's inequality.

Solution:

(i) Let $k = 2\log(n)$. We will show that with a very high probability the size of each $T[i]$ will be at most k. For $i \in [0, n-1]$, let E_i be the event that the size of $T[i]$ is at least k.

Note that event E_i occurs if there is a subset of S of size k that is mapped to i. Thus,

$$Prob(E_i) \;\leqslant\; {}^nC_k\Big(\frac{1}{n}\Big)^k \;\leqslant\; \frac{1}{k!} \;\leqslant\; \frac{1}{(k/2)^{(k/2)}}$$

Now, let $E = \cup_{i=0}^{n-1} E_i$ be the event that for at least one i, size of $T[i]$ is at least k.

By union bound,

$$Prob(E) \leqslant \sum_{i=0}^{n-1} Prob(E_i) \;\leqslant\; n \cdot \frac{1}{(k/2)^{(k/2)}} \;\leqslant\; \frac{1}{n^5}\;.$$

Now let $X = \big(\max_{i=0}^{n-1}\; T[i]\big)$. Note that the maximum value of X is n. The expected size of X can be calculated as follows:

$$Exp(X) \;\leqslant\; n \cdot Prob(E) \;+\; 2\log n \cdot Prob(E^c) \;\leqslant\; n \cdot \frac{1}{n^5} \;+\; 2\log n \;=\; O(\log n)\;.$$

(ii) The expected value of maximum time taken to verify the membership of elements of U in S $Exp\big(\max_{i=0}^{n-1}\; T[i]\big)$ which by part (i) is $O(\log n)$.

(iii) We have $H(x)$ as $x(\bmod n^2)$, for $x \in U$. For any $i, j \in S$, define a random variable Y_{ij} as follows.

$$Y_{ij} \;=\begin{cases} 1 & \text{if } H(i) = H(j); \\ 0 & \text{otherwise.} \end{cases}$$

Further, let $Y = \sum_{\substack{i,j\in S \\ i\neq j}} Y_{ij}$. Then Y denotes the number of collisions. Now,

$$Exp(Y) \;=\; \sum_{\substack{i,j\in S \\ i\neq j}} Exp(Y_{ij}) \;=\; \sum_{\substack{i,j\in S \\ i\neq j}} Prob(Y_{ij}=1) \;=\; \sum_{\substack{i,j\in S \\ i\neq j}} \frac{1}{n^2} \;\leqslant\; \frac{1}{2}\;.$$

Observe that Y is a non-negative random variable. So by Markov's inequality $Prob(Y \geqslant 1)$ is bounded by $Exp(Y)$ which in turn is bounded above by $1/2$.

This proves that with probability at least half Y is 0 (that is, there are no collisions).

Quiz 2

Question 1 Let $X, Y, Z \in \{0,1\}^n$ be three n-length strings. Describe an $O(n^2)$ time algorithm to compute largest k such that there exists a k-length string that is substring of X, Y, and Z.

Solution: For $i = 1$ to n perform the following two steps.

1. Set pattern $P = X[i, n]$.
   - Compute a table A of size n such that $A[j]$ stores the length of largest suffix of $Y[1, j]$ that is prefix of P.
   - Compute a table B of size n such that $B[j]$ stores the length of largest suffix of $Z[1, j]$ that is prefix of P.

   (It was proved in Lecture 14 that tables A, B are computable in linear time.)

2. Sort the elements of A, B in linear time (using bucket sort), and find the maximal common entry (say m_i).

   Claim: The length of largest prefix of P that lies in both Y, Z is m_i.
   Proof: See Lecture 14.

After performing n iterations, we return the answer as $\max_{i=1}^{n}(m_i)$.

Question 2 Let $G = (V, E)$ be a weighted digraph with no cycle of negative weight, and let $S \subseteq V$ be a set of size k. A path P is said to be an S-path if the internal vertices of P lie in S.

Describe an $O(kn^2)$ time algorithm to compute a binary matrix B such that $B[i, j] = 1$ if and only if there exists an S-path of negative weight from vertex i to vertex j in G.

Solution:

1. Without loss of generality assume $1, \ldots, k$ are elements of S (if not, rearrange vertices in V).
2. Create matrix D of size $n \times n$ as follows:

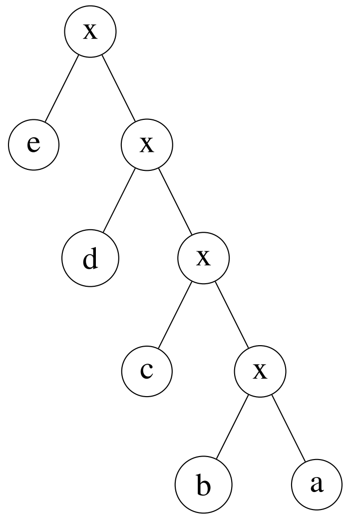
$$D[i,j] = \begin{cases} 0 & \text{if } i = j; \\ wt(i,j) & \text{if } (i,j) \in E; \\ \infty & \text{otherwise.} \end{cases}$$

3. For $k = 1$ to n:
   For $i, j = 1$ to n:
   $D[i, j] = \min\{D[i, j], D[i, k] + D[k, j]\}$
4. Compute another matrix B such that $B[i, j] = 1$ iff $D[i, j] < 0$.
5. Return B.

Quiz 1

Question 1 Compute the optimal prefix free encoding for a character set consisting of 5 letters $\{a, b, c, d, e\}$ with frequencies as: $a : 1,\ b : 1,\ c : 2,\ d : 3,\ e : 5.$

Solution: The tree corresponding to prefix free encoding will be:



Question 2 Prove or disprove the following statement: "Any n vertex graph G with $n-1$ bridges has a unique MST".

Solution: Consider any MST T of G. Note that none of the edges outside T can be a bridge edge. As there are $n-1$ bridges, each edge of T must be a bridge, which in turn proves that $T = G$. Thus, the MST must be unique.

Question 3 Let $G = (V, E)$ be a DAG and let $s \in V$ be a vertex in G. Design an optimal algorithm to verify that there is a unique path (at most one path) between all pairs in $\{s\} \times V$, and analyse its time complexity.

Solution: Compute a DFS tree T of G with respect to node s, and in the process if encountered with a forward/cross edge then return "Not a unique path".

The time complexity is $O(n)$ since we scan at most n edges.

The correctness follows from the fact that a forward/cross edge (say (a, b)) results in two distinct s to b paths in G.