Problem Set 3

Justin Ely

Foundations of Algorithms
July 5, 2016
612-240-0924

1a)

With weights of $[5, 9, 10, 2]$ and $k = 10$, the number of trucks used will be 4, where the minimum possible would be 3 given a more efficient algorithm.

1b)

After playing with this for a while, and listening to office hours, I still wasn't able to come up with an adequate solution. The office hours recording referenced variables that were seemingly undefined, and some of the assumptions made didn't seem to fit in the problem definition.

Nevertheless, I tried a bunch of scenarioes and always found the best and worst cases to be within a factor of 2 of each-other, so it does seem to be the correct result.

2a)

The probability of any new node to link to a specific existing node is $P_l = \frac{1}{k-1}$. With a node v_j, where only nodes added after j can possibly have linked to it, the expected number of linked nodes is going to be the sum of the probabilites of each new node:

$$\begin{aligned} N_l &= \sum_{k=j+1}^{n} \frac{1}{k-1} && (1)\\ &= \sum_{k=j}^{n-1} \frac{1}{k} && (2)\\ &= \int_0^{n-1} \frac{1}{k} - \int_0^{j} \frac{1}{k} && (3)\\ &= ln(n-1) - ln(j) && (4)\\ &= ln(\frac{n-1}{j}) && (5)\\ &= \theta(ln(\frac{n}{j})) && (6) \end{aligned}$$

2b)

The probability of each new node to not link to a specific existing node is $P = 1 - \frac{1}{k-1}$. Thus, the expectation value is given by:

$$\begin{aligned} E &= \prod_{k=j+1}^{n} (1 - \frac{1}{k-1}) && (7)\\ &= \frac{j-1}{j}\frac{j}{j+1}\frac{j+1}{j+2}...\frac{n-2}{n-1} && (8)\\ &= \frac{j-1}{n-1} && (9) \end{aligned}$$

Thus the total number of unlinked nodes is:

$$\begin{aligned} N &= \sum_{j=1}^{n} \frac{j-1}{n-1} && (10)\\ &= \frac{1}{n}\sum_{j=1}^{n} j-1 && (11)\\ &= \frac{1}{n-1}\frac{n(n-1)}{2} && (12)\\ &= \frac{n}{2} && (13) \end{aligned}$$

3a)

Pop operations run in $O(1)$ time, and in worst case MultiPopA, MultiPopB, and Transfer will need to exhaust the full stack. This gives worst case running times of:

- MultiPopA: $O(n)$
- MultiPopB: $O(m)$
- Transfer: $O(n)$

3b)

Define $\phi(n,m) = n + m$ and $\hat{c_i} = \phi(D_{i+1}) - \phi(D_i)$, thus:

PushA

$$\begin{aligned}
\hat{c_i} &= c_i + \phi(D_{i+1}) - \phi(D_i) && (14)\\
&= 1 + (n+1) + m - n + m && (15)\\
&= 1 + 1 && (16)\\
&= 2 && (17)
\end{aligned}$$

PushB

$$\begin{aligned}
\hat{c_i} &= c_i + \phi(D_{i+1}) - \phi(D_i) && (18)\\
&= 1 + n + (m+1) - n + m && (19)\\
&= 1 + 1 && (20)\\
&= 2 && (21)
\end{aligned}$$

MultiPopA

$$\begin{aligned}
\hat{c_i} &= c_i + \phi(D_{i+1}) - \phi(D_i) && (22)\\
&= k + (n-k) + m - n + m && (23)\\
&= k - k && (24)\\
&= 0 && (25)
\end{aligned}$$

MultiPopB

$$\begin{aligned}
\hat{c_i} &= c_i + \phi(D_{i+1}) - \phi(D_i) && (26)\\
&= k + n + (m-k) - n + m && (27)\\
&= k - k && (28)\\
&= 0 && (29)
\end{aligned}$$

Transfer

$$\begin{aligned}
\hat{c_i} &= c_i + \phi(D_{i+1}) - \phi(D_i) && (30) \\
&= 2k + (n - k) + (m + k) - n + m && (31) \\
&= 2k + 0 && (32) \\
&= 2k && (33)
\end{aligned}$$

We see from the amortized cost of each operation that they each run in constant time: $O(1)$.