15-351 / 15-650 / 02-613 (Fall 2019): Midterm #2 Rubric

1. **Short answer. (i-v: 4 pts per question, vi: 5pts)**

   (i) By ascendant or descendant order. The question is asking for order, only specifying the running time would lead to lose of points.

   (ii) $\Omega(n) \sim O(n)$. No need to be perfect skip list.

   (iii) Root of the tree. Please check the slides for details.

   (iv) Increasing order of $j-i$. Writing down the the bellman equation without stating the order would lead to lose of points. Explaining that we solve the problem by solving sub-problems gains no point since it is too high level.

   (v) The maximum value of the network flow in this problem is n, and you will have $2n+m$ edges in this graph, so the runtime is $\mathbf{O}(n(2n+m))$, you will also get the full grade if you write $\mathbf{O}(nm)$.

   (vi) By induction on $|A|$.

   If $|A| = 1$, that is only node s in A, the claim is correct by definition.

   Assume the claim is true when $|A| \leq k$, that is $v(f) = f^{out}(A) - f^{in}(A)$ when $|A| \leq k$. Let $A' = A \cup \{x\}$ and $B' = B - x$, x is some node in the graph. Denote $f(A', B')$ be the net flow from A' to B', we have:

   $$f(A', B') = f^{out}(A') - f^{in}(A') = f(A, B') + f(x, B')$$

   and

   $$f(A, B) = f^{out}(A) - f^{in}(A) = f(A, B') + f(A, x) = f(A, B') + f(x, B')$$

   The last equation is because the node except source and sink should have the property that flow-in is equal to flow-out.

   Then we have:

   $$\begin{aligned} v(f) &= f^{out}(A) - f^{in}(A) = f(A, B) \\ &= f(A, B') + f(x, B') = f(A', B') - f(x, B') + f(x, B') \\ &= f(A', B') = f^{out}(A') - f^{in}(A') \end{aligned}$$

2. residual graph: 15 pts, max-flow graph: 5pts, minimum cut: 5pts

   Please see Figure 1. Note that the answer of the max flow graph is not unique.

3. $DP(i, j)$ calculates the length of longest common subsequence between $s_1[0:i]$ and $s_2[0:j]$. $(0 \leq i < |s_1|,\ 0 \leq i < |s_2|)$

   $$DP(i, j) = \max \begin{cases} DP(i-1, j-1) + 1 & \text{if } s_1[i] == s_2[j] \\ \max\{DP(i-1, j),\ DP(i, j-1)\} & \text{otherwise} \end{cases}$$

   $DP(|s_1| - 1, |s_2| - 1)$ stores the length of the longest common subsequence.

   To get the sequence, trace back in the DP matrix from entry $(|s_1| - 1, |s_2| - 1)$. If $s_1[i] == s_2[j]$, walk diagonally. Otherwise, walk to $\text{argmax}\{DP(i-1, j),\ DP(i, j-1)\}$.

(Correct recurrence = 15pt. Every mistake in recurrence may lead to deduction of 1 or 2 points.)

Basecase(5pt): $DP(0,j) = DP(i,0) = 0$ for all i,j.

Runtime(5pt): The algorithm runs in $O(n^2)$ as it fills up the matrix. Tracing back takes $O(\max\{|s_1|,|s_2|\})$ time.

4.

$$OPT(j,k) = \max \begin{cases} OPT(j-1,k) & \text{if } p_j \notin \text{output} \\ OPT(j-1,k-t(p_j)) + q(p_j) & \text{otherwise} \end{cases}$$

(Correct recurrence = 15 pts. Every mistake in recurrence may lead to deduction of 1 or 2 points.)

Basecase (5 pts): $OPT(0,k) = DP(j,0) = 0$ for all j,k.

Runtime (5 pts): The algorithm runs in $O(nk)$.

Extra (15 pts): Solve the exam design problem for each category / concept separately. Then combine them together for the final results. Takes $O(nkc)$ time.

Assignment Project Exam Help

https://powcoder.com

Add WeChat powcoder

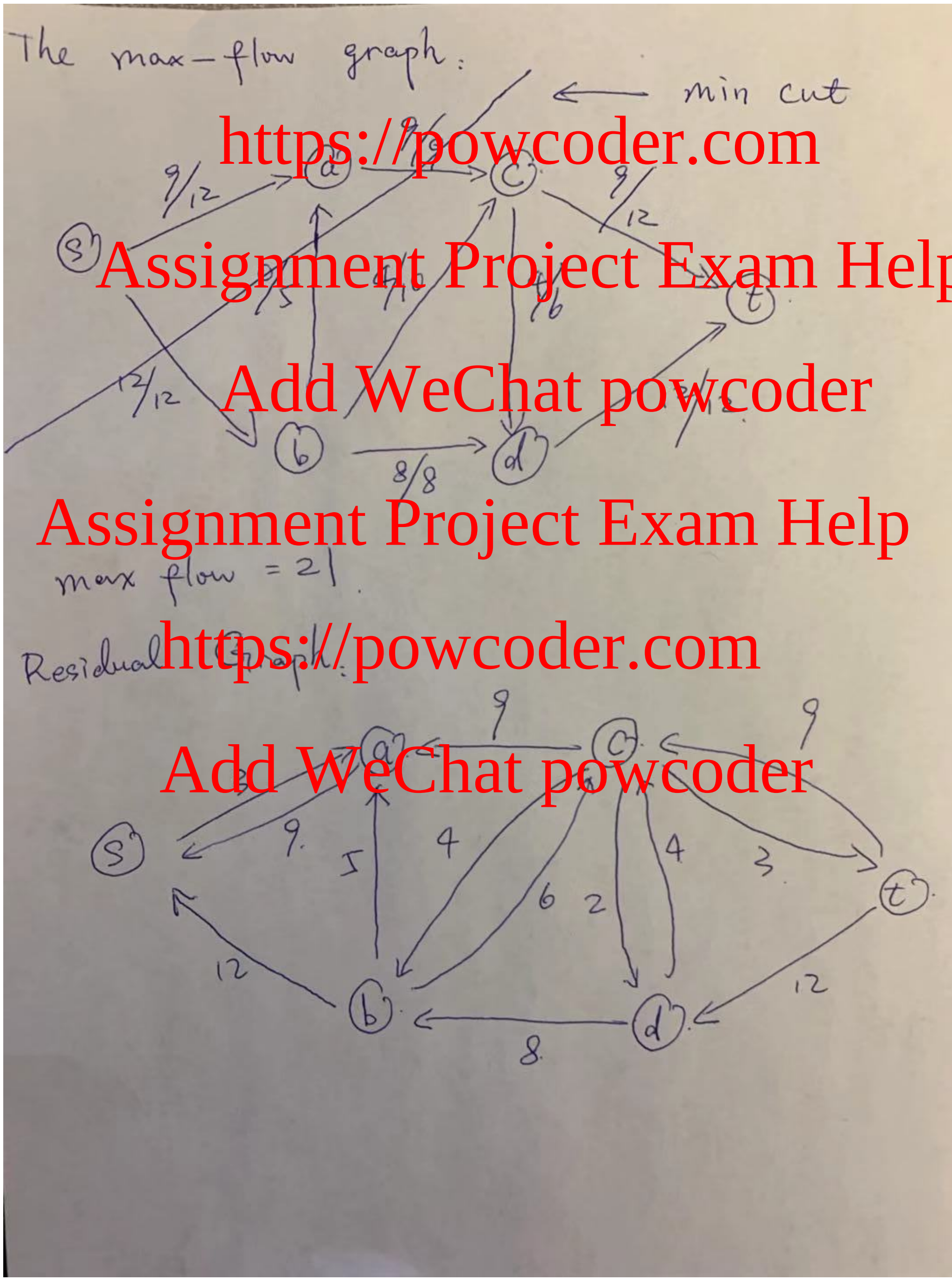

Figure 1: Answer of the problem 2