NPTEL Questions

CP — July-Nov 2021

Week 3 — Greedy Algorithms

1. (1 point) It's festive season and you want to give away some gifts over n days. At first, you have nothing. On the i-th day, a_i gifts are delivered by Amazon. You can give away at most 8 of these gifts and keep the remaining to give away later. What is the minimum day index at the end of which you would have given away k gifts? The days are indexed from 1 to n. For example, if $n = 3$, $k = 17$, and $a_1 = a_2 = a_3 = 10$, then the answer is 3. If $n = 1$, $k = 10$ but $a_i = 9$, then you won't be able to give away a total of k gifts over n days, and in this case the answer can be given as -1.

   Suppose k is `167` and n is `99`, with the a_i's given by:

   ```
   72 4 79 73 49 58 15 13 92 92 42 36 35 21 13 10 51 94 64 35 86 50 6 80 93 77 59 71 2 88
   22 10 27 30 87 12 77 6 34 56 31 67 78 84 36 27 15 15 12 56 80 7 56 14 10 9 14 59 15 20
   34 81 8 49 51 72 4 58 38 77 31 86 18 61 27 86 95 36 46 36 39 18 78 39 48 37 71 12 51 92
   65 48 39 22 16 87 4 5 42
   ```

   You can also find the numbers listed above at this URL:

   `https://raw.githubusercontent.com/neeldhara/NPTEL-CP-2021/main/Week02-AssignmentP1`

   What is the minimum day index at the end of which you would have given away k gifts for this example?

   Expected answer: 21

   Explanation. Let t be denote the number of gifts we have. On the i-th day, we increase t by a_i ,then we give away $\min(8, t)$ gifts, and decrease k by this value. Use this idea to determine the final answer.

2. (2 points) Recall the `Island Wars` problem discussed in the lectures:

   There are N islands lining up from west to east, connected by $N-1$ bridges. The i-th bridge connects the i-th island from the west and the $(i+1)$-th island from the west. One day, disputes took place between some islands, and there were M requests from the inhabitants of the islands:

   Request i: A dispute took place between the a_i-th island from the west and the b_i-th island from the west. Please make traveling between these islands with bridges impossible.

   You decided to remove some bridges to meet all these M requests. The goal was to find the minimum number of bridges that must be removed.

   (a) How many bridges need to be destroyed in the example given at the following URL?

   `https://raw.githubusercontent.com/neeldhara/NPTEL-CP-2021/main/Week02-AssignmentP2`

   The format of the input is exactly as discussed in class.

   Expected answer: 306

   Explanation: Run the code described in class on this example.

   (b) What is the relationship between:

   x = maximum # of disjoint requests, and

   y = minimum # of bridges that need to be destroyed to serve all requests?

   Infer as much as you can.

   A. $x \leq y$

   B. $x \geq y$

   C. $x = y$

   D. No particular relationship is guaranteed in all cases.

   Explanation: Note that y is clearly at least x; but by employing the strategy described in class, all requests can be handled by destroying at most x bridges, so y is also at most x.

3. (4 points) There are **n** dragon heads and **m** knights. Each dragon head has a *diameter* and each knight has a *height*. A dragon head with diameter D can be chopped off by a knight with height H if $D \leq H$. A knight can only chop one dragon head. Asking a knight k to chop any dragon head will incur a cost of $h(k)$ gold coins, where $h(k)$ denotes the height of the knight k.

   Given a list of diameters of the dragon heads and a list of heights of the knights, we want to know if it is possible to chop off all the dragon heads. If yes, we would like to determine a manner of chopping all dragon heads with minimum total cost. In other words, we want to find the **minimum** total heights of the knights used to chop the dragon heads.

   (a) Consider the following strategies to solve this problem.

   A. Create a biparitite graph G with vertices corresponding to knights and dragon heads, and add an edge between a knight u and a head v if u can chop v of weight $h(u)$. Find a maximum matching M of minimum cost in G. If the size of the matching is less than n, then the answer is `not feasible`, otherwise it is the cost of the matching M.

   B. Match each dragon head d to a knight with the shortest height that is at least as tall as the diameter of d. If each dragon head finds a match, then the answer is the heights of the matched knights. Otherwise, the answer is `not feasible`.

   C. For each dragon head d, find the knight with the shortest height that is at least as tall as d. If there is no such knight, the answer is `not feasible`. Otherwise, eliminate the dragon head d and the corresponding knight from the instance, and move to the next dragon head.

   D. For each dragon head d, find the knight with the tallest height that is at least as tall as d. If there is no such knight, the answer is `not feasible`. Otherwise, eliminate the dragon head d and the corresponding knight from the instance, and move to the next dragon head.

   Which of the strategies above constitute a valid approach to the problem?

   A. all of the above
   B. (A) and (C)
   C. (A), (B), and (C)
   D. only (A)
   E. (A) and (D)
   F. (B), (C), and (D)

   Explanation: The solution, when it exists, corresponds to a matching in the graph G of size n. The costs on the edges ensure that we have a solution with optimal cost as well. The second approach fails to remove matched knights, so one dragon head may get matched with multiple knights. The correctness of the third option can be shown with a greedy stays-ahead argument. The last approach fails on every feasible example with more than one dragon head: all dragon heads get matched to the tallest knight!

   (b) For each dragon head d, how much time does it take to find the knight with the shortest height that is at least as tall as d? Assume that the knights are sorted according to their height.

   A. $O(n \log n)$ B. $O(m \log m)$ C. $O(\log n)$ **D.** $O(\log m)$ E. $O(mn)$

   Explanation: Use (slightly modified) binary search.

   (c) For each dragon head d, how much time does it take to find the knight with the shortest height that is at least as tall as d? Assume that the heights of the knights are stored in a C++ `multiset`.

   A. $O(n \log n)$ B. $O(m \log m)$ C. $O(\log n)$ **D.** $O(\log m)$ E. $O(mn)$

   Explanation: Retrieving an element from a multiset requires $O(\log s)$ time from a set with s elements.

   (d) Suppose we are using approach (C) from above on an instance where n and m can be as large as 10^5. If we use a linear search to find a suitable knight for each dragon head, and the time limit on the judge is two seconds, what response are we likely to get?

   A. WA **B. TLE** C. AC D. PE E. None of the above

   Explanation: This approach will require $O(nm)$ time, corresponding to approximately 10^{10} operations in the worst-case, and the solution is likely to time out.

4. (3 points) The following questions concern the stable marriage problem discussed in class.

   (a) In a stable marriage instance, if a man M and woman W each put each other at the top of their respective preference lists, then M must be paired with W in every stable matching.

   A. True B. False

   Explanation: In a stable marriage instance, if a man M and woman W each put each other at the top of their respective preference lists, then M must be paired with W in every stable matching.

   (b) In every instance of the Stable Matching Problem, there is a stable matching containing a pair (M, W) such that M and W are each others' favorites - in other words, M is ranked first on the preference list of W and W is ranked first on the preference list of M.

   A. True **B. False**

   This is false, since even the input preferences may not have such a pair. As a concrete example, consider the group m_1, m_2, w_1, and w_2 with the following preferences.

   - m_1 has the ranking: $w_1 \succ w_2$
   - m_2 has the ranking: $w_2 \succ w_1$
   - w_1 has the ranking: $m_2 \succ m_1$
   - w_2 has the ranking: $m_1 \succ m_2$

   It can be easily checked that in this situation, it's impossible to have a pair (m, w) such that m is ranked first on the preference list of w and w is ranked first on the preference list of m.

   (c) Suppose that all of the men rank the women in the same order, and all of the women also rank the men in the same order. Then, there is exactly one stable matching.

   A. True B. False

   Explanation. Indeed, there is only one stable matching in this setting. Let there be n people of each sex and number the men and women 1 through n in order of their rank by the opposite sex. The one and only stable matching has woman i match with man i for all $i = 1, ..., n$.

   To prove this claim we need to show two things:

   1. This matching is stable.
   2. No other matching is stable.

   Stability: Suppose that man i and woman i are matched for each i. Suppose that two people of opposite sexes prefer each other to their partner in the original match. Since this partnership is not the i to i matching, one of these two people must have a higher rank than the other. The person with the higher rank is currently matched to a partner of his or her own rank. Therefore he or she prefers his or her current partner to the partner in the proposed alternative match. Thus there can be no two people who prefer each other to their assigned partners.

   Uniqueness: Consider a matching where not everyone is of the same rank as his or her partner. Then at least there is at least one pair of persons of the same rank who are not matched to each other. Let j be the rank of the two highest-ranked persons who are not matched with each other. Since this is the highest-ranked pair who are not matched together, it must be that each member of the pair is matched with someone of lower rank. This means that man j would prefer woman j to his current partner and woman j would prefer man j to her current partner. So the proposed matching is not stable. This proves that if everyone ranks the opposite sex in the same way, then no matching in which persons are not matched with someone of their own rank is stable.

   Food for thought: if all men rank women the same way, but women have different rankings for the men, is it STILL true that there is only one stable matching?