CS205: Design and analysis of algorithms

End-semester examination • Maximum marks: 40 • Spring 2023

Notes

- Instructor/TA will not be available for clearing doubts during the exam.
- If any assumptions are made, state them and justify.
- There are 5 questions. Each question carries 7 marks, except Q3, which carries 12 (10+2) marks.
- You can use the results we obtained in the lectures, without repeating the proofs discussed in the lectures.
- Hand-written materials are allowed, but no printed materials and electronic gadgets are allowed.
- Write legibly, or risk losing marks.

Q1: Use Strassen's algorithm to compute the matrix product AB, where $A = \begin{pmatrix} 1 & 3 \\ 7 & 5 \end{pmatrix}$ and $B = \begin{pmatrix} 6 & 8 \\ 4 & 2 \end{pmatrix}$.

Sketch of Answer: $S_1 = [6], S_2 = [4], S_3 = [12], S_4 = [-2], S_5 = [6], S_6 = [8], S_7 = [-2], S_8 = [6], S_9 = [-6], S_{10} = [14]$.
$P_1 = [6], P_2 = [8], P_3 = [72], P_4 = [-10], P_5 = [48], P_6 = [-12], P_7 = [-84]$.
$C_{11} = [18], C_{12} = [14], C_{21} = [62], C_{22} = [66]$.

Evaluation: There are 21 product/summation that you have to calculate. Correctly finding each of them gives you 1/3 marks. Note that both the correct value and correct application of the formula are important.

Q2: Derive Huffman code for the string "acatisacatindeed".

Sketch of Answer: One of the answer is with the following codewords: $a : 11, c : 010, t : 011, i : 100, s : 0010, n : 0011, d : 101, e : 000$.

The symbol a has frequency 4 and it gets a codeword of size 2. The symbols c, t, i, d, e are having frequencies 2 each and they get a codeword of length 3 each. The symbols s, n are having frequencies 1 each and they get codewords of length 4 each.

Evaluation: There are 8 symbols. If you get the size of the codeword of a symbol wrong, you loose 1 mark. Of course, if you are wrong for all symbols you get 0 marks (not -1). The steps are also important. For example, directly multiplying the two matrices will not give you any credit.

Q3: Let G be a flow network shown in Figure 1.

(a) Find a maximum flow in G using Ford-Fulkerson method.

(b) Find the cut of G corresponding to the flow obtained by you.

Sketch of Answer: The maximum flow possible is 33. A cut corresponding to this flow is $S = \{s, a, b, c, d, e, f, g\}, T = \{h, t\}$.

Evaluation: If you get (by applying Ford-Fulkerson method) a flow of x, then you get marks $10x/33$. If you correctly find the cut for the flow you obtained (S contains all the vertices reachable from s in the last residual network and T contains the rest of the vertices), you get 2 marks. Since maximum flow is equal to the capacity of the minimum cut, if you find a suboptimal flow, whatever cut you find will be wrong. So, the second part is relevant to only those who score full for the first part.

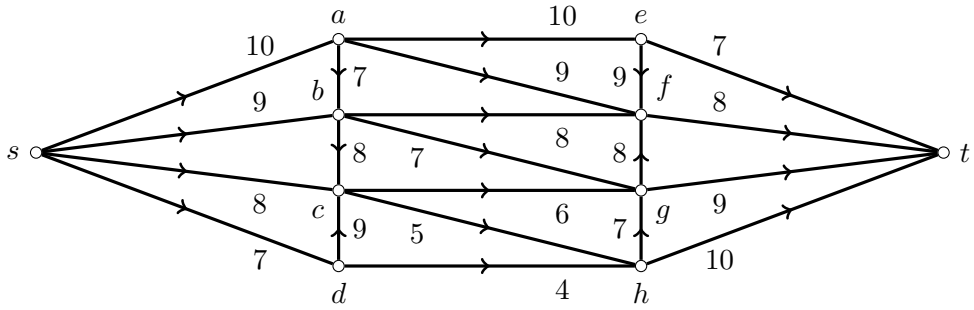

Figure 1

Q4: Consider Algorithm 1 for the optimization version of INDEPENDENT SET problem. Prove or disprove that it is a 2-approximation algorithm.

Algorithm 1 Algorithm for INDEPENDENT SET

```
1: procedure ALG-IS(G)
2:     Initialize I = ∅.
3:     while G has at least one vertex do
4:         Find a minimum degree vertex v in G.
5:         I = I ∪ {v}.
6:         Remove v and all its neighbors from G.
7:     end while
8:     Return I
9: end procedure
```

Sketch of Answer: It is not a 2-approximation algorithm. For example, consider the following graph G with 11 vertices: $V = \{v\} \cup I(= \{i_1, i_2, i_3, i_4, i_5\}) \cup C(= \{c_1, c_2, c_3, c_4, c_5\})$. The set I forms an independent set and the set C forms a clique. The vertex v is adjacent to all vertices in I but not adjacent to vertices in C. Every vertex in I is adjacent to every vertex in C. The size of the maximum independent set in G is 5 (the unique maximum independent set in G is I). Note that v is the minimum degree vertex (degree 5) in G. So, the algorithm picks v and delete v and all its neighbors (I) from G. What remains is C from which the algorithm picks one vertex, say c_1. There are no vertices remaining after removing c_1 and all its neighbors (all other vertices in C). So, the set returned by the algorithm has size 2.

Evaluation: If you are trying to prove the statement you score 0 marks. Marks will be given for finding a correct counter example.

Q5: Prove or disprove: Every optimal prefix code is an Huffman code.

Sketch of Answer: Not every optimal prefix code is an Huffman code. For example, consider a string with frequencies of symbols as following: $a : 3, b : 3, c : 2, d : 2$. In every Huffman code all the four symbols get codewords of size 2 each. But each Huffman code, the starting bit of both a and b are the same. Therefore, $a : 00, b : 10, c : 01, d : 11$ is an optimal prefix code but not an Huffman code.

Evaluaton: If you are trying to prove the statement, you score 0 marks. Marks will be given to correct counter-examples.