CS 170 HW 12

Due **2021-4-27, at 10:00 pm**

1 Study Group

List the names and SIDs of the members in your study group. If you have no collaborators, write "none".

2 Local Search for Max Cut

Sometimes, local search algorithms can give good approximations to NP-hard problems. In the Max-Cut problem, we have a graph $G(V, E)$ and we want to find a cut (S, T) with as many edges crossing as possible. One local search algorithm is as follows: Start with any cut, and while there is some vertex $v \in S$ such that more edges cross $(S - v, T + v)$ (or some $v \in T$ such that more edges cross $(S + v, T - v)$), move v to the other side of the cut. Note that when we move v from S to T, v must have more neighbors in S than T.

(a) Give an upper bound on the number of iterations this algorithm can run for (i.e. the total number of times we move a vertex).

(b) Show that when this algorithm terminates, it finds a cut where at least half the edges in the graph cross the cut.

3 Coffee Shops

A rectangular city is divided into a grid of $m \times n$ blocks. You would like to set up coffee shops so that for every block in the city, either there is a coffee shop within the block or there is one in a neighboring block. (There are up to 4 neighboring blocks for every block). It costs r_{ij} to rent space for a coffee shop in block ij.

Write an integer linear program to determine which blocks to set up the coffee shops at, so as to minimize the total rental costs.

(a) What are your variables, and what do they mean?

(b) What is the objective function?

(c) What are the constraints?

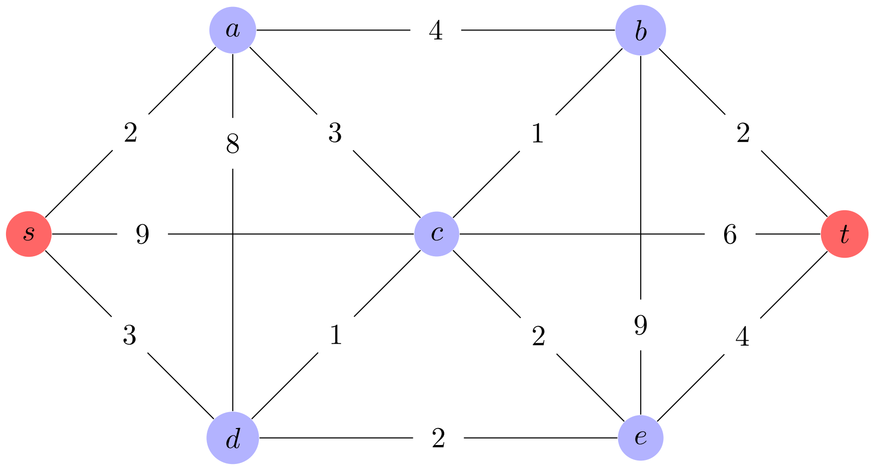
(d) Solving the linear program gets you a real-valued solution. How would you round the LP solution to obtain an integer solution to the problem? Describe the algorithm in at most two sentences.

(e) What is the approximation ratio obtained by your algorithm?

(f) Briefly justify the approximation ratio.

4 Mechanical Project Problem

In this problem, you will look at a few mechanical examples of the project problem, to build intuition. Consider the graph $G = (V, E)$ below.



Your answers to the following *do not* require any justification.

(a) What is the shortest path from s to t in G and what is its length?

(b) Which edge should be removed to maximize the length of the shortest path if you are allowed to remove $k = 1$ edge? What is the new shortest path length from s to t?

(c) Which edges should be removed to maximize the length of the shortest path if you are allowed to remove $k = 2$ edges? What is the new shortest path length from s to t?

(d) Suppose that instead of being allowed to remove individual edges, you are allowed to remove exactly one node (not s or t) from the graph. Which node should be removed to maximize the length of the shortest path? What is the new shortest path length from s to t?

5 Reduction Coding (Extra Credit)

This semester, we are trying something new: questions which involve coding.

This link will take you to a python notebook, hosted on the Berkeley datahub, in which you will implement a reduction from the 3-coloring problem to 3SAT, and then use the provided SAT solver, along with your reduction, to solve 3-coloring problems. Once you have finished, download a PDF of your completed notebook via File → Download as → PDF via HTML, and append the downloaded pdf to the rest of your homework submission. Be careful when selecting pages on gradescope.

Note: Datahub does not guarantee 100% reliability when you save your notebook, and recommends downloading a local copy occasionally to backup progress (via File → Download as → Notebook (.ipynb)).