Exam

written test, 2 hours

2 parts:

1) theory questions — 3 (~4 points each)
2) problem solving — 2 (~10 points each)

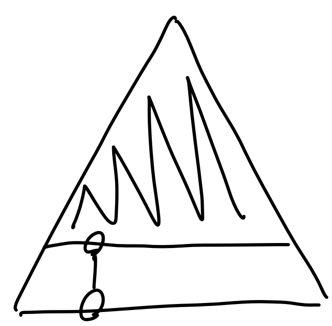
~32 points

see Moodle for examples

Exercise (approx. alg.): (alternative 2-approx for Vertex Cover)

Consider the following algorithm for Vertex Cover:

- run DFS from an arbitrary vertex of G
- return all the non-leaf vertices of the DFS tree



1) Show that this algorithms returns a vertex cover of G

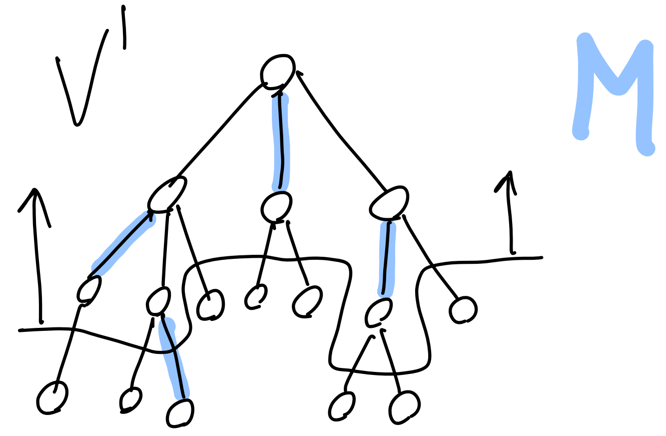
2) Show that this algorithm is a 2-approx algorithm for Vertex Cover
(Hint: show a large enough matching in the DFS tree)

3) Show a lower bound of 2 to the approximation factor of this algorithm

Solution:

1) The parents of the leaves cover all the edges left uncovered by the leaves of the DFS tree

2)



Let r be the root, choose one child v and add (r, v) to the matching Π. $\forall$ level $i \geq 1$ consider all vertices V not endpoints of any edge of Π; choose a child u and add (V, u) to Π. Repeat up to the leaves

a) upper bound to the cost of V'

By construction, M matches all the vertices of V', and since each of such edges has at most 2 endpoints in V'

$$|V'| \leq 2|M|$$

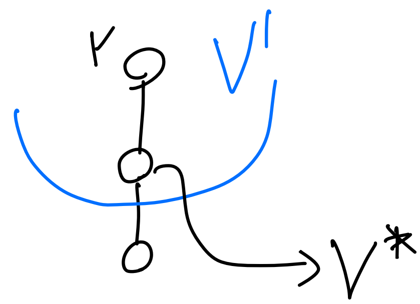
b) lower bound to the cost of $V^* = OPT$

$\forall$ matching M of G, $|V^*| \geq |M|$

(M matching $\Rightarrow$ in any vertex cover, in particular V^*, there must be ≥ 1 vertex $\forall$ edge of M) (seen before in this course)

$$\Rightarrow |V'| \leq \cancel{2|M|} \leq 2|V^*|$$

3) Show that the 2-factor is tight:



i.e: star graph, with DFS starting from a leaf

Single-linkage clustering

Clustering: given a set X of n data points, partition them into "coherent groups" (called clusters) of "similar points"

similarity function f: assigns to each pair of data points a real number that specifies their "similarity". smallest f = most similar

Goal: a K-clustering = partition data points into K non-empty clusters

Single-linkage clustering: at the beginning every data point is in its own cluster; then successively merge the two clusters containing the most similar pair of points belonging to $\neq$ clusters, until K clusters remain

Exercise: Give a fast implementation of single-linkage clustering

obs.: hey, this is Kruskal's algorithm!
↳ stopped early

1) Define a complete graph $G = (X, E)$ with vertex set X and one edge $(x,y) \in E$ of weight $f(x,y)$ for each pair of vertices $x, y \in X$

2) Run Kruskal's alg. on G until the solution T contains $n-k$ edges (or, equivalently, until k connected components remain)

3) Compute the connected components of (X, T) and return the corresponding partition of X

Complexity: $O(n^2 \log n)$