1. Consider the following algorithm for the MAX-CUT problem.

A. Start with an arbitrary cut (S,T) of V.

B. So long as possible, repeat:

   (i) If there exists u ∈ S such that the cut (S–u,T+u) has more cross edges than (S,T), delete u from S, and add u to T.
   (ii) If there exists v ∈ T such that the cut (S+v,T–v) has more cross edges than (S,T), add v to S, and delete v from T.

C. Return (S,T).

$O(mn^2)$

Cut size keeps on increasing

(a) Prove that this algorithm terminates in polynomial time.

(b) Prove that the approximation ratio of this algorithm is 1/2.

u $\quad b_u = \#$ of nbrs of u in its part

$c_u = \#$ of nbrs of u in the other part

$b_u + c_u = d_u$

Algo terminates $\Rightarrow$ $c_u \geq b_u \ \forall u$

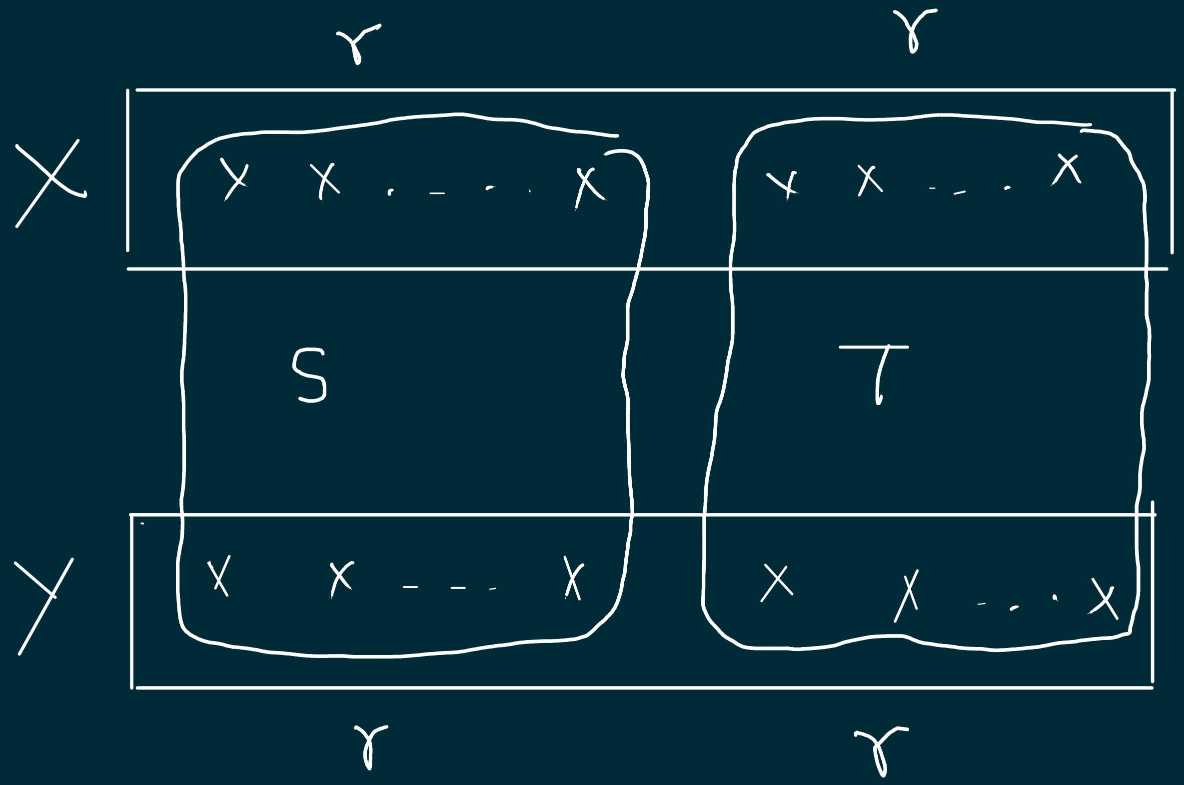
$2c_u \geq c_u + b_u = d_u$

$\frac{|c(S,T)|}{OPT} \geq 1/2$ $\qquad 2\sum_u c_u \geq \sum_u d_u = 2m$

$2|c(S,T)| = \sum_u c_u \geq m \geq OPT$

(c) Prove that this approximation ratio is tight.

Supply an <u>infinite</u> family of examples



$K_{2r,2r}$

S, T init
final

$$|c(S,T)| = 2r^2$$

$$|E| = 4r^2$$

2. Consider the following algorithm for EUCLIDEAN-TSP.

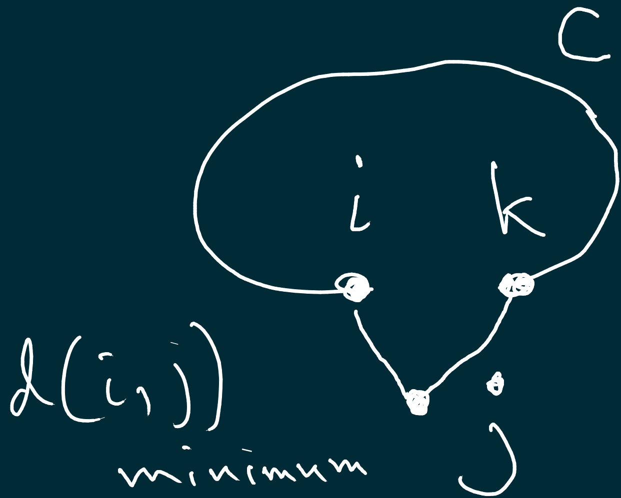
a) Select the pair (u,v) such that d(u,v) is smallest among all pairs.
Start with the tour C = (u,v).

n-2 times

b) Repeat until C is a Hamiltonian cycle:

Find i ∈ C and j ∉ C such that d(i,j) is the minimum. Let k be the city next to i in C. Replace i,k by i,j,k in C.

Prove that this is a 2-approximation algorithm.



The algorithm runs for $n-2$ iterations. Let the t-th iteration replace i_t, k_t by i_t, j_t, k_t. The initial edge (u, v) and the edges (i_t, j_t) for $t = 1, 2, \ldots, n-2$ construct a minimum spanning tree of the complete weighted graph on n vertices (the cities), because Prim's algorithm chooses precisely these edges.

At the beginning, the 2-city tour has cost $d(u,v) + d(v,u) = 2d(u,v)$. The t-th iteration increases the cost of the tour by the amount $\delta_t = d(i_t, j_t) + d(j_t, k_t) - d(i_t, k_t)$. By the triangle inequality, we have $d(j_t, k_t) \leqslant d(j_t, i_t) + d(i_t, k_t)$, that is, $d(j_t, k_t) \leqslant d(i_t, j_t) + d(i_t, k_t)$, that is, $d(j_t, k_t) - d(i_t, k_t) \leqslant d(i_t, j_t)$. Therefore $\delta_t \leqslant 2d(i_t, j_t)$. The cost of the final tour produced by the algorithm is

$$c = 2d(u,v) + \sum_{t=1}^{n-2} \delta_t \leqslant 2\left[d(u,v) + \sum_{t=1}^{n-2} d(i_t, j_t)\right],$$

where the sum within square brackets is the cost of the MST.

Let OPT be the cost of the optimal tour. If we delete any edge e from the optimal tour, we get a spanning tree of cost $\mathrm{OPT} - d(e)$. We clearly have

$$\mathrm{OPT} \geqslant \mathrm{OPT} - d(e) \geqslant \mathrm{cost}(\mathrm{MST}),$$

that is,

$$c \leqslant 2 \times \mathrm{cost}(\mathrm{MST}) \leqslant 2 \times \mathrm{OPT}.$$

3. [Next-fit strategy for BIN-PACKING] Keep on adding the items to the most recently opened bin so long as possible. When the insertion of the next item lets the capacity of the current bin exceed, close the current bin, and open a new bin.

(a) Prove that this is a 2-approximation algorithm.

(a) Let m be the number of bins used by the next-fit strategy. Suppose that a bin j is at most half full in this packing. But then, the next bin $j+1$ must be more than half full. Otherwise, the opening of bin $j+1$ is not justified. More importantly, the total weight of the two bins j and $j+1$ must be larger than the bin capacity C. In fact, the sum of the weights of the items placed in bin j and the weight of the first item put in bin $j+1$ must be larger than C. It follows that among the first $m-1$ bins, at most $m/2$ bins can be at most half full, each followed by a bin more than half full that makes the average weight of the two consecutive bins larger than $C/2$. Therefore the total weight packed is $\sum_{i=1}^{n} a_i > Cm/2$. Moreover, $C \times \mathrm{OPT} \geqslant \sum_{i=1}^{n} a_i$, that is, $m < 2 \times \mathrm{OPT}$. Since m and OPT are integers, we have $m \leqslant 2 \times \mathrm{OPT} - 1$.

(b) Prove that the approximation ratio of 2 is tight, that is, given any $\varepsilon > 0$, there exists a collection for which the approximation ratio is $> 2 - \varepsilon$.

(b) Given $\varepsilon > 0$, choose a positive integer $k > \frac{1}{\varepsilon}$. Let there be $k-1$ items of weight k, and k items of weight 1, and $C = k$. Suppose that these items occur in the sequence $1, k, 1, k, \ldots, 1, k, 1$. We have $\mathrm{OPT} = k$ (place all the objects of weight 1 in one bin, and each item of weight k in a single bin; this strategy must be optimal since all the k bins are filled to the capacity). The next-fit algorithm uses $m = 2k - 1$ bins. The approximation ratio is therefore $2 - \frac{1}{k} > 2 - \varepsilon$.