Honour Code: No Internet Searches, No collaboration!

1. (15 points) **(Grocery Store Packing)** You are given n items each with some known weight $0 \leq w_i \leq 1$, and a bunch of bags. Each bag has a weight tolerance of 1 kg beyond which it will break. The goal is to put the items into bags to minimize the number of bags used. Give a simple 2-approximation to this problem, i.e., come up with an algorithm which uses at most twice the optimal number of bags. Does your approximation ratio improve if the weight of each item is known to be at most some $\alpha < 1$.

   Solution:

   Proof. $1 + 1 = 2$. □

2. (20 points) **(Guessing the Largest Number)** Here is a combined question on online and stochastic algorithms. I (the adversary) have chosen a set U of n distinct real numbers (arbitrary). But the nice thing is that I reveal them to you in a completely random permuted form (uniformly out of the $n!$ permutations). Your goal, upon seeing these numbers in an online manner, is to decide which of them is the largest number of the set U. For example, when you see the first element, you must either declare it to be the largest number of U, or move on to the second element. Once you move on, you can not go back in time and declare some early element to be the largest.

   (a) (10 points) Consider the following algorithm: suppose you see the first $n/2$ numbers, and let the maximum among them be v. In the next $n/2$ numbers, declare the first one *greater than* v to be the largest. If there is none, declare the last element to be the largest element of U. Show that this algorithm works with decent probability. What lower bound can you prove on the probability with which you will be correct.

   Solution:

   Proof. $1 + 1 = 2$. □

   (b) (10 points) If you optimize the threshold of how many numbers you see (instead of $n/2$, try $c \cdot n$ for some c), what threshold maximizes the correctness probability?

   Solution:

   Proof. $1 + 1 = 2$. □

3. (25 points) **(Of Flows and Cuts with Several Origin-Destination pairs)** You should recall the famous *max-flow min-cut theorem* which ascertains the following beautiful theorem: given a undirected graph $G = (V, E)$ where each edge has unit capacity, and a source vertex s and sink vertex t, consider the two quantities: Let $F_{\max}$ denote the

maximum flow which s can send to t while respecting edge capacities; let $C_{\min}$ denote the *s-t min-cut*, i.e., the minimum number of edges crossing any partition of the form $(S, V \setminus S)$ where $s \in S$ and $t \in V \setminus S$. Then $F_{\max} = C_{\min}$. In fact, one such proof of this theorem is by using LP duality. In this question, we will study the same from a more practical setting.

In this question, there are many origin-destination requests of the form (s_1, t_1), (s_2, t_2), $\ldots$, (s_k, t_k) where s_1 wants to send some flow to t_1, s_2 wants to send some flow to t_2, and so on. The goal is to maximize the minimum throughput, i.e., send some λ flow from all s_i to t_i such that λ is maximized.

We now state below, a simple linear program which captures this problem. The set $P_{(s_i,t_i)}$ is the set of all simple paths from s_i to t_i, i.e., every element $p \in P_{(s_i,t_i)}$ is a simple path from s_i to t_i. Ignore the fact that there are exponentially many variables — we are not trying to implement this in a computer, we are only using LPs to understand/prove some very fundamental mathematical properties about flows and cuts in this question. In fact, this exercise shows how to use algorithmic techniques like LPs and Duality to show deep mathematical properties.

$$
\begin{aligned}
&\max \lambda \\
&\text{s.t} \sum_{p \in P_{(s_i,t_i)}} f_p \geq \lambda && \forall 1 \leq i \leq k \\
&\sum_i \sum_{p \in P_{(s_i,t_i)} \text{ s.t } e \in p} f_p \leq 1 && \forall e \in E \\
&f_p \geq 0 && \forall i, \forall p \in P_{(s_i,t_i)}
\end{aligned}
$$

(a) (5 points) Let $F_{\max}$ then denote the maximum flow, i.e., optimum value of the above LP. Likewise, analogous to $C_{\min}$ defined above, we can define a similar cut value: consider partitioning the graph vertices into disjoint sets S and $\overline{S}$. Like done in class, define the sparsity of this cut as the ratio of $\frac{|E(S,\overline{S}|}{N(S,\overline{S})}$ where $N(S, \overline{S})$ is the number of origin-destination pairs separated by the cut. Let $C_{\min}$ denote the smallest sparsity over all possible cuts. Show that $F_{\max} \leq C_{\min}$.

Solution:

Proof. $1 + 1 = 2$. □

(b) (10 points) Consider the following graph in Figure 1, and let the origin-destination pairs be as follows: $s_1 = a$ and $t_1 = b$, $s_2 = b$ and $t_2 = c$, $s_3 = a$ and $t_3 = c$, $s_4 = u$ and $t_4 = v$. In this example, calculate the sparsest cut (show that no sparsest cut exists), and also show that $F_{\max}$ must be strictly smaller than $C_{\min}$. This shows that having more than one O-D pair makes the max flow-min cut theorem false!

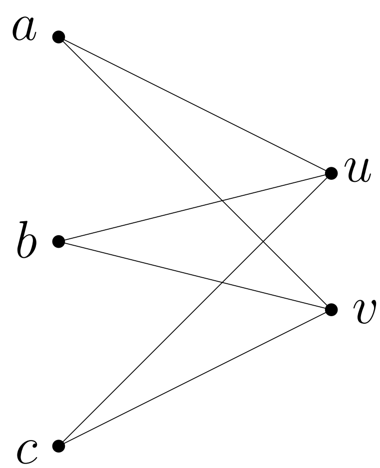

Figure 1: Figure for Gap Example

Solution:

Proof. $1 + 1 = 2$. □

(c) (10 points) Write the dual of the LP above, and say why it is a natural *LP relaxation* of the problem of finding the sparsest cut defined above in part (a).

Solution:

Proof. $1 + 1 = 2$. □

(d) (10 points) Show that the optimal dual solution can be viewed as a metric $d(,\cdot,)$ (like discussed in class) satisfying $d(s_i, t_i) \geq 1$ for all s_i and t_i. Using the ℓ_1 embedding theorem and ideas discussed in class and in HW4, find a cut of sparsity at most $O(\log n)$ times the LP value of the dual. Using this, show the following approximate max-flow min-cut theorem: $\frac{C_{\min}}{O(\log n)} \leq F_{\max} \leq C_{\min}$. So even though the equality does not hold, we have used LP-duality and LP-rounding to prove an approximate equality.