Assignment 1 Solution

Maxime Descoteaux

February 3, 2003

Question 1 (25 pts)

Hereditary Property (6 pts)

1) Sushi problem

Suppose B is an admissble set. Let A $\subseteq$ B (any subset of B)
Assume for a contradiction that A is inadmissible. Then, elements in A cost more than b yen. But since A $\subseteq$ B, sushi in B must also cost more than b yen, CONTRADICTION! Thus, A must be admissible.

2) Matchings

Suppose B is an admissble set. Let A $\subseteq$ B.
Assume for a contradiction that A is inadmissible. Then, there exists at least two edges e_1 and e_2 in A that share a common vertex. But, e_1 and e_2 are also in B, which means B is inadmissible as well. CONTRADICTION! Thus, A must be admissible.

3) Committee

Suppose B is an admissble set. Let A $\subseteq$ B. This means that $|A| \leq |B| \leq k$. Thus, A is asmissible as well.

4) Independent Columns

Suppose B is an admissble set. Let A $\subseteq$ B.
Assume for a contradiction that A is inadmissible. Then, there are some indexe d columns in A that are linearly dependent. But these columns are also in set B. Thus, B is inadmissble as well. CONTRADICTION! Thus, A must be admissible.

5) TSP

Suppose B is an admissble set. Let A $\subseteq$ B.
If A = B then the property is satisfied trivially because B is admissible. If $|A| < |B|$ then we know that by adding all edges in B-A (set difference) to A, we get back the set B which is itself extendable or already a cycle. Hence, A is also admissible.

6) Forest

Suppose B is an admissble set. Let A $\subseteq$ B.
Assume for a contradiction that A is inadmissible. Then, there exist at least one cycle in A. But this means the cycle is also in set B. Thus, B is inadmissible as well. CONTRADICTION! Thus, A must be admissible.

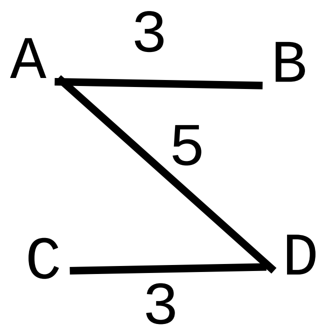

Figure 1: matching

Exchange Property

1) Sushi

i)Find an instance where greedy fails (1 pt)
ii)Counter example to show exchange property fails (2 pt).
b = 10 yen
Sushis = (kamikaze, 10 yen, 8 score), (sunshine, 5 yen, 5 score), (bubinga, 4 yen, 4 score)
Greedy choice: kamikaze $\implies$ cost = 10 yen , score = 8
Maximal choice: sunshine, bubinga $\implies$ cost = 9 yen , score = 9
Thus, greedy fails!
Let A = {kamikazee} and B = {sunshine, bubinga}. We cannot add any sushi from B to A since we will exceed the 10 yen we have to spend. Hence, exchange property fails.

2) Matchings

i)Find an instance where greedy fails (1 pt)
ii)Counter example to show exchange property fails (2 pt).
Consider Figure1. Greedy chooses AD with score = 5 but maximal solution is {AB, CD} with score = 6. Thus, greedy fails.
Let A = {AD} and B = {AB, CD}. No edge from B can be added to A without having two edges sharing a common vertex. Hence, exchange property fails.

3) Committee

i)Proof of echange property (3 pts)
Suppose A and B are admissible and that $|A| < |B|$. Now, assume for a contradiction that there exists no x in B that can be added to A such that $A \cup \{x\}$ is admissible. Then, this means $|A| \geq k$. But B is a strictly bigger set than A, which means $|B| > k$. So, B is inadmissible. CONTRADICTION!
Therefore, the exhange property must be satisfied.

4) Linear Independence

i)Proof of exchange property (3 pts)
Suppose A and B are admissible and that $|A| = k$ and $|B| = l$, where $k < l$. Now, assume for a contradiction that we cannot add any column in B-A to A so that $A \cup \{x\}$ is admissible. This means that all columns in B can be written as a linear combination of the k columns in A. From Linear Algebra, this means that the rank(B) $\leq$ k. So, this either implies that $|B| \leq |A|$ or that B is linearly dependent which means it

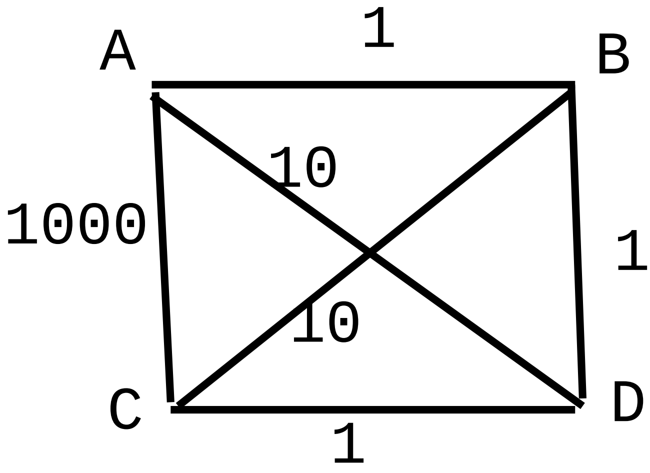

Figure 2: tsp

is inadmissible. In both cases, we have a CONTRADICTION! Hence, the exchange property must be satisfied.

5) TSP

i)Find an instance where greedy fails (2 pt)
ii)Counter example to show exchange property fails (2 pt).
Consider Figure2.
Greedy choice: AB, BD, DC, CA with score = 1003
Minimum choice: AD, DC, CB, BA with score = 22
Greedy fails!
Now, let A = {DC, CB, BA} and B = { AB, BD, DC, CA}. A and B are both admissible since A can be extended into a cycle and B is already a valid tour. But, adding any edge in B to A will not complete a valid tour. Hence, exchange property fails!

6) Forest

ii)Proof of exchange property (3 pts)
Suppose A and B are admissible and that $|A| < |B|$. Assume for a contradiction that we cannot add any edge x in B-A to A so that $A \cup \{x\}$ is admissible. This means that $A \cup \{x\}$ creates a cycle **for all** x in B. You should convince yourself that this implies that $|B| \leq |A|$ or that B itself contains a cycle. A rigorous proof can be done by induction. In both cases, we have a CONTRADICTION! Hence, the exchange property is satisfied.

Question 2 (20 pts)

i)Algorithm (idea, details, implementation) (10 pts)
ii)Complexity (5 pts)
iii)proof of correctness (5 pts)

Key idea: Suppose we are given a graph G and a valid spanning tree. The importance of this question was to note that adding any edge not in G-T to T was **creating a cycle**. Now, if there exists a smaller weight edge e in that cycle then we know that T is not minimal. Adding e to T creates a cycle but deleting any bigger weight edge of that cycle still maintains the spanning tree properties while producing a strictly less weighted tree. Hence, for all edge in G-T, we repeat this procedure.

implementation and algorithm: Straight forward from description above. The only subtlety is how to find cycles when adding e from G-T to T. We can use a Depth First Search approach to do so.

complexity:
-main loop runs in worst case: $|E| - (|V| - 1)$ times
-finding a cycle using depth first search. O($|V|$)
-going around the cycle. In worst-case, the cycle is $|V| - 1$ long.
Overall: O($|V||E|$)

proof:
First, as noted before, no matter what our algorithm does, we know we are returning a spanning tree. But does it return a strictly smaller tree in the case where T is not a MST or does it answer "yes we have a MST" in the case T is a MST?
case1: suppose given T is not a MST
Suppose the return tree T' is not strictly less than T. Then, this means that we have returned the same tree T or one which weighs more. If we have returned the same tree, it means that all edge in G-T have smaller weight than all edges in the cycles they create when added to T. This means T was already a MST. CONTRADICTION! In the other case, if we have returned a larger weighted tree T', we also have a CONTRADICTION! since our algorithm only deletes edges in T by smaller weighted edges.
case2: suppose given T is a MST
Then, by definition, any edge in G-T has bigger or equal weight to any edge in the cycle it creates when adding it to T. Hence, our algrothim will never find a strictly smaller weight edge in any cycle produced by adding any edge of G-T to T. Therefore, the algorithm does not modify T and returns "yes we have a MST".
End of proof!

Note:
1) Looking only at adjacent edges of all vertex in T is NOT enough. Must look for entire cycles. Look at Figure3.
2)I took off half marks for algorithms running Kruskals's or Prim's eventhough correct. This was not needed and was not the point of the exercise. 3)Always give an outline of your algo in words. Looking at pseudo-code does not tell me much and is frustrating to read.

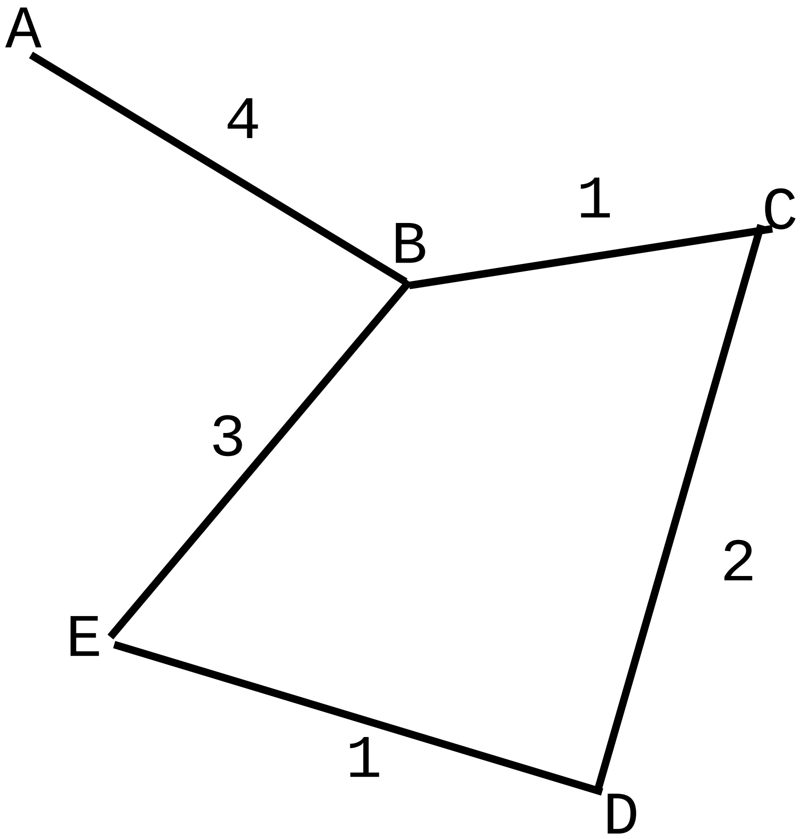

Figure 3: Suppose you are given T = {AB, BC, BE, ED}. The correct algorithm adds edge CD to T and notices that this creates a cycle (CD, DE, EB, BC) where edge EB has strictly larger weight than CD. Thus, you remove it from T and add CD instead. However, many of you were only looking at adjacent edges to vertices in T. This will never replace edge BE by DC because neighbors of edge DC are already in T and have lesser weight and similarly for BE.