CSCI 5421 - Assignment 7

Email	neis@umn.edu
Name:	Kelsey Neis

16.4-1

Show that (S, I_k) is a matroid where S is any finite set and I_k is the set of all subsets of S of size at most k, where $k <= |S|$

1. S is a **finite** set: true by problem statement
2. **Hereditary**:
    1. $B_k \subseteq S \; such \; that \; k <= |S|$
    2. Then, for any $A_{k'} \subseteq B_k, \; k' <= k$
    3. $A_{k'} \in I_k$ because $I_k = all \; subsets \; of \; S$ size k, which includes $all \; subsets \; of \; B$
3. **Exchange property**:
    1. $A \in I_k \; and \; B \in I, \; |B| > |A|$
    2. Suppose $\nexists \; x \in B, \; \notin \; A \; and \; \{x\} \bigcup A \; \in \; I_k$
    3. Then there is some $A' \notin I_k$ which contains all elements from subsets within I_k
    4. Since $B \in \; I_k$ and I_k contains all possible subsets of size k, this leads to a contradiction
    5. Thus, $\{x\} \bigcup A \in I_k$ and the exchange property is met, completing the proof that (S, I_k) is a matroid.

16.4-2

Given an $m \times n$ matrix T over some field (such as the reals), show that (S, I) is a matroid, where S is the set of columns of T and $A \in I$ if and only if the columns in A are linearly independent.

1. S is **finite**: T is over the real numbers, and is therefore finite. It follows that S is finite as well.
2. **Hereditary**:

   Sufficient:

   1. If $A \in I$, and assuming the columns of A are linearly independent, then there is no set of coefficients, other than the trivial coefficients 0, such that the columns of A add up to zero: $c_1\vec{v_1} + c_2\vec{v_2} + ... + c_i\vec{v_i} = \vec{0}$. So, in this case, $c_1 = c_2 = ... = c_i = 0$.
   2. Given (1), if $B \subseteq A$, then checking for linear independence means taking 0 or more elements from the above equation. This would still result in all zero coefficients, so B must also be linearly independent.

   Necessary:

   1. If, on the other hand, we assume A's columns are *not* linearly independent, then I cannot be, since $A \in I$ and the dependency in A would also be contained in I, and therefore (S, I) would not be a matroid.
3. **Exchange property**:

   Sufficient:

   1. $A \in I \ and \ B \in I, \ |B| > |A|$.
   2. Suppose $\nexists \ \vec{v} \in B, \ \notin \ A \ and \ \{\vec{v}\} \bigcup A \ \in \ I$.
   3. This implies that A is not extensible, meaning that it has the maximum number of columns for linear independence to be maintained.
   4. However, if A is not extensible, then $B \notin I$, since $|B| > |A|$, which contradicts (1). Therefore there must be a $\vec{v} \in B, \notin A$ such that $\{\vec{v}\} \bigcup A \in I$.

   Necessary:

   1. If we do not assume A is linearly independent, then there cannot be a $\vec{v}$ that when added to A, makes the set independent, because a dependency is already present.

16.4-4

> ***16.4-4*** ⋆
> Let S be a finite set and let $S_1, S_2, \ldots, S_k$ be a partition of S into nonempty disjoint subsets. Define the structure $(S, \mathcal{I})$ by the condition that $\mathcal{I} = \{A : |A \cap S_i| \leq 1$ for $i = 1, 2, \ldots, k\}$. Show that $(S, \mathcal{I})$ is a matroid. That is, the set of all sets A that contain at most one member of each subset in the partition determines the independent sets of a matroid.

1. **Finite**: S is finite by the problem statement.
2. **Hereditary** (using B and C here not to confuse with A from the definition of I):
    1. $B \in I$ and $C \subseteq B$
    2. B in I means that B must contain at most 1 member of each subset in the partition, following the definition of I.
    3. C being a subset of B means that it also contains at most 1 member of each subset in the partition, since taking away 0 or more members from B to form C cannot change this property. Thus I is hereditary.
3. **Exchange property:**
    1. If $C, B \in I$ and $|C| < |B|$, then there must be one member of B which belongs to a subset in the partition which C has no member in.
        1. Adding it to C would therefore not break the condition for being in I - to at most contain one member from the subsets of the partition.
        2. If this were not the case, then $|C| < |B|$ would imply that B contained more than one member from one of the subsets, contradicting $B \in I$
    2. This means that there exists an $x \in B \backslash C$ such that $C \bigcup \{x\} \in I$ and it satisfies the exchange property.

16.4-5

Show how to transform the weight function of a weighted matroid problem, where the desired optimal solution is a *minimum-weight* maximal independent subset, to make it a standard weighted-matroid problem. Argue carefully that your transformation is correct.

The minimum weight maximal independent subset is similar to the maximum weight problem in that all maximal independent subsets will be of the same size. So, to transform the weight function, we need to invert the order of the weights. That way, the greedy algorithm on p. 440 can be used to find the minimum weight maximal independent subset. By finding the maximum sum of inverted weights which maintains independence, a minimum subset is found:

1. Set w_0 to a number greater than the maximum length for any edge
2. $w_{min}(A) = \sum_{e \in a} w_0 - w(e)$

Proof:

We have $w_0 - w(min) = max_i(w_0 - w_i)$

Therefore, the greedy algorithm will choose weights in ascending order.

17.3-1

Suppose we have a potential function Φ such that $\Phi(D_i) \geq \Phi(D_0)$ for all i, but $\Phi(D_0) \neq 0$. Show that there exists a potential function Φ' such that $\Phi'(D_0) = 0$, $\Phi'(D_i) \geq 0$ for all $i \geq 1$, and the amortized costs using Φ' are the same as the amortized costs using Φ.

Solution:

For all i in n, $\Phi'(D_i) = \Phi(D_i) - \Phi(D_0)$

Proof that amortized costs are the same:

Substituting $\Phi'(D_i)$ into equation 17.3, we get:

$\sum_{i=1}^{n} \hat{c}_i' = \sum_{i=1}^{n}(c_i + (\Phi(D_i) - \Phi(D_0)) - (\Phi(D_{i-1}) - \Phi(D_0))$

The $\Phi(D_0)$ terms cancel out:

$\sum_{i=1}^{n} \hat{c}_i' = \sum_{i=1}^{n}(c_i + \Phi(D_i - \Phi(D_{i-1}))$

$= \sum_{i=1}^{n} \hat{c}_i' = \sum_{i=1}^{n} c_i + \Phi(D_n) - \Phi(D_0)$
$= \sum_{i=1}^{n} \hat{c}_i$

17.3-3

> Consider an ordinary binary min-heap data structure with n elements supporting the instructions INSERT and EXTRACT-MIN in $O(\lg n)$ worst-case time. Give a potential function Φ such that the amortized cost of INSERT is $O(\lg n)$ and the amortized cost of EXTRACT-MIN is $O(1)$, and show that it works.

The real costs of performing Insert and Extract-Min operations come from maintaining the min-heap property. In both cases, the potential is a function of the depth of the heap

Set $\Phi(D_n) = \sum_{i=1}^{n} lg\ i = lg(n!) <= nlgn$

Insert:

$\hat{c}_i = c_i + \Phi(D_{after}) - \Phi(D_{before})$

$= lgn + nlgn - (n-1)lg(n-1)$

$= lgn + nlgn - nlg(n-1) + lg(n-1)$

$= lgn + nlg(\frac{n}{n-1})$

For sufficiently large n, $nlg(\frac{n}{n-1}) < lg\ n$

$\rightarrow$ O(lgn)

Extract-Min:

$\sum_{i=1}^{k} \hat{c}_i = \sum_{i=1}^{k} c_i + \Phi(D_i) - \Phi(D_{i-1})$

Where $\Phi(D_i)$ is the function of the state of the min-heap, i.e. the cost of Extract-Min, *after* the operation and $\Phi(D_{i-1})$ is the function of the state of the min-heap *before* the operation:

$\Phi(D_{after}) = (n-1)lg(n-1)$

$\Phi(D_{before}) = nlg\ n$

$\hat{c}_i = c_i + \Phi(D_{after}) - \Phi(D_{before})$

$= lgn + (n-1)lg(n-1) - nlg\ n$

$= lgn + nlg(n-1) - lg(n-1) - nlg\ n$

$= lg(\frac{n}{n-1}) + nlg(\frac{n-1}{n}) \propto 1$

$\rightarrow \mathrm{O}(1)$

17.3-5

Suppose that a counter begins at a number with b 1s in its binary representation, rather than at 0. Show that the cost of performing n INCREMENT operations is $O(n)$ if $n = \Omega(b)$. (Do not assume that b is constant.)

$\sum_{i=1}^{n} c_i = \sum_{i=1}^{n} \hat{c_i} - \Phi(D_n) + \Phi(D_0)$

(given an amortized cost of $\hat{c_i} = 2$)

$<= \sum_{i=1}^{n} 2 - b_n + b$

$= 2n - b_n + b$

Since $b, b_n <= n$, the total cost remains to be proportional to n, thus the cost of performing n Increment operations is $\mathrm{O}(n)$

17.4-3

Suppose that instead of contracting a table by halving its size when its load factor drops below 1/4, we contract it by multiplying its size by 2/3 when its load factor drops below 1/3. Using the potential function

$$\Phi(T) = |2 \cdot T.num - T.size| \ ,$$

show that the amortized cost of a TABLE-DELETE that uses this strategy is bounded above by a constant.

Case 1: $\frac{1}{3} < size <= size$

$num_{after} = num_{before} - 1$

$size_{after} = size_{before}$

$\hat{c}_{remove} = 1 + (2num_{after} - size_{after}) - (2num_{before} - size_{before})$

$= 1 + 2(num_{before} - 1) - 2num_{before}$

$= -1$

Case 2: $size <= \frac{1}{3}$

$num_{before} = \frac{1}{3}size_{before}$

$num_{after} = num_{before} - 1$

$size_{after} = \frac{2}{3}size_{before}$

$\hat{c}_{remove} = \frac{1}{3}size_{before} + 1 + (\frac{1}{2}size_{after} - num_{after}) - (\frac{2}{3}size_{before} - num_{before})$

$= 2$