

CSE 6140 / CX 4140

Computational Science & Engineering Algorithms

Homework 4

Please type in all answers.

1. (25 points)

A store trying to analyze the behavior of its customers will often maintain a two-dimensional array A , where the rows correspond to its customers and the columns correspond to the products it sells. The entry $A[i, j]$ specifies the quantity of product j that has been purchased by customer i .

Here's a tiny example of such an array A .

	liquid detergent	beer	diapers	cat litter
Raj	0	6	0	3
Alanis	2	3	0	0
Chelsea	0	0	0	7

One thing that a store might want to do with this data is the following. Let us say that a subset S of the customers is *diverse* if no two of the of the customers in S have ever bought the same product (i.e., for each product, at most one of the customers in S has ever bought it). A diverse set of customers can be useful, for example, as a target pool for market research.

We can now define the Diverse Subset Problem as follows: Given an $m \times n$ array A as defined above, and a number $k \leq m$, is there a subset of at least k of customers that is *diverse*?

Show that Diverse Subset is NP-complete.

Hint: Independent Set is NP-complete.

2. (25 points)

Suppose you're helping to organize a summer sports camp, and the following problem comes up. The camp is supposed to have at least one counselor who's skilled at each of the n sports covered by the camp (baseball, volleyball, and so on). They have received job applications from m potential counselors. For each of the n sports, there is some subset of the m applicants qualified in that sport. The question is: For a given number $k < m$, is it possible to hire at most k of the counselors and have at least one counselor qualified in each of the n sports? We'll call this the *Efficient Recruiting Problem*.

Show that Efficient Recruiting is NP-complete.

Hint: Vertex Cover is NP-complete.

3. (25 points)

A convoy of ships arrives at a port and delivers a total of n containers, each containing a different kind of hazardous material. Waiting near the port is a set of m trucks, each of which can hold up to k containers. Any container can be placed in any truck; however, there are certain pairs of containers that cannot be placed together in the same truck. The chemicals they contain may react explosively if brought into contact.

The *Truck Loading Problem* is as follows. Is there a way to load all n containers into the m trucks so that no truck is overloaded, and no two containers are placed in the same truck when they are not supposed to be?

Show that Truck Loading is NP-complete.

Hint: 3-Coloring is NP-complete.

4. (25 points)

The mapping of genomes involves a large array of difficult computational problems. At the most basic level, each of an organism's chromosomes can be viewed as an extremely long string (generally containing millions of symbols) over the four-letter alphabet $\{A, C, G, T\}$. One family of approaches to genome mapping is to generate a large number of short, overlapping snippets from a chromosome, and then to infer the full long string representing the chromosome from this set of overlapping substrings.

While we won't go into these string assembly problems in full detail, here's a simplified problem that suggests some of the computational difficulty one encounters in this area. Suppose we have a set $S = \{s_1, s_2, \dots, s_n\}$ of short DNA strings over a q -letter alphabet; and each string s_i has length 2ℓ , for some number $\ell \geq 1$. We also have a library of additional strings $T = \{t_1, t_2, \dots, t_m\}$ over the same alphabet; each of these also has length 2ℓ . In trying to assess whether the string s_b might come directly after the string s_a in the chromosome, we will look to see whether the library T contains a string t_k so that the first ℓ symbols in t_k are equal to the last ℓ symbols in s_a , and the last ℓ symbols in t_k are equal to the first ℓ symbols in s_b . If this is possible, we will say that t_k *corroborates* the pair (s_a, s_b) . (In other words, t_k could be a snippet of DNA that straddled the region in which s_b directly followed s_a .)

Now we'd like to concatenate all the strings in S in some order, one after the other with no overlaps, so that each consecutive pair is corroborated by some string in the library T . That is, we'd like to order the strings in S as $s_{i_1}, s_{i_2}, \dots, s_{i_n}$, where i_1, i_2, \dots, i_n is a permutation of $\{1, 2, \dots, n\}$, so that for each $j = 1, 2, \dots, n - 1$, there is a string t_k that corroborates the pair $(s_{i_j}, s_{i_{j+1}})$. (The same string t_k can be used for more than one consecutive pair in the concatenation.) If this is possible, we will say that the set S has a *perfect assembly*.

Given sets S and T , the *Perfect Assembly Problem* asks: Does S have a perfect assembly with respect to T ? Prove that Perfect Assembly is NP-complete.

Example. Suppose the alphabet is $\{A, C, G, T\}$, the set $S = \{AG, TC, TA\}$, and the set $T = \{AC, CA, GC, GT\}$ (so each string has length $2\ell = 2$). Then the answer to this instance of Perfect Assembly is yes: We can concatenate the three strings in S in the order $TACAGTA$ (so $s_{i_1} = s_2$, $s_{i_2} = s_1$, and $s_{i_3} = s_3$). In this order, the pair (s_{i_1}, s_{i_2}) is corroborated by the string CA in the library T , and the pair (s_{i_2}, s_{i_3}) is corroborated by the string GT in the library T .

Hint: Hamiltonian Path is NP-complete.