The set cover problem

Exercise: Set cover

This exercise asks you to implement an algorithm that computes a low-cost set cover. More precisely, the input consists of a nonempty family $\mathcal{F} = \{S_0, S_1, \ldots, S_{m-1}\}$ of nonempty subsets $S_j \subseteq \{0, 1, \ldots, n-1\}$ for $j = 0, 1, \ldots, m-1$. A set $R \subseteq \{0, 1, \ldots, m-1\}$ is a *set cover* if for all $x \in \{0, 1, \ldots, n-1\}$ there exists a $j \in R$ with $x \in S_j$. The *cost* of a set cover R is $c(R) = |R|$. The *optimum cost* OPT of a set cover is the minimum cost of a set cover, where the minimum is taken over all possible set covers; if no set cover exists, the optimum is undefined. The algorithm must either

> 1. output a set cover R with cost $c(R) \le H_q \cdot \mathrm{OPT}$, where $q = \max_{j=0,1,\ldots,m-1} |S_j|$ and $H_q = 1 + 2 + \ldots + \frac{1}{q}$, or
> 2. correctly assert that no set cover exists.

For example, an implementation of the greedy set cover algorithm suffices for this purpose.

Your task in this exercise is to complete the subroutine

```
void solver(int n, int m, const int *p, const int *f, int &k, int *r)
```

which should compute the size `k` and the elements `r` of a set cover as described in the previous paragraph from the given input consisting of positive integers `n` and `m`, as well as the arrays `p` and `f`, whose format is as follows.

The array `f` concatenates the sets $S_0, S_1, \ldots, S_{m-1}$. That is, writing $S_j[0], S_j[1], \ldots, S_j[|S_j|-1]$ for the $|S_j|$ distinct elements of the set S_j, we have $f = (S_0[0], S_0[1], \ldots, S_0[|S_0|-1], \ldots, S_{m-1}[0], S_{m-1}[1], \ldots, S_{m-1}[|S_{m-1}|-1])$.

The array `p` satisfies $p[i] = \sum_{0 \le j < i} |S_j|$ for all $i = 0, 1, \ldots, m$. Thus, the array `p` provides an index to the array `f` with $S_j = \{f[p[j]], f[p[j]+1], \ldots, f[p[j+1]-1]\}$ for all $j = 0, 1, \ldots, m-1$.

For example, when $m = 2$, $S_0 = \{4, 6, 8\}$, and $S_1 = \{7, 9\}$, we have $f = (4, 6, 8, 7, 9)$ and $p = (0, 3, 5)$.

The output of the subroutine should be as follows. To give as output a set cover $R = \{j_0, j_1, \ldots, j_{k-1}\}$, set `k` equal to k and the element `r[i]` equal to j_i for all $i = 0, 1, \ldots, k-1$. When no set cover exists, set `k` equal to 0. You may assume that $1 \le n \le 1048576$, $0 \le m \le 2097152$, $0 \le p[m] \le 16777216$, $0 \le k \le n$, $q \le 10$, and that the array r has capacity for at least n elements. To locate the subroutine quickly, you can search for "`???`" in the source file.

Grading. This exercise awards you up to 10 points in the course grading. The number of points awarded is the maximum points times the number of tests passed over the total number of tests, rounded up. To successfully complete a test, your implementation must use no more than 10 seconds of wall clock time and 1 GiB of memory. Each test will in general require the successful solution of one or more problem instances. In each batch of scaling tests, the first failed test will cause all subsequent tests in the batch to be skipped.

Points **10 / 10** My submissions **1 / 10**
Deadline Tuesday, 8 November 2022, 20:00
To be submitted alone

This course has been archived (Friday, 30 December 2022, 16:00).

Set cover

1. Download the template source file set-cover.cpp.
2. Study the problem description above.
3. Complete your solution in the template.
4. Test your solution with the unit tests available in the template.
   Recall that the Scratchpad is available for compiling and running your solution.
5. Submit your solution for grading using the dialog below.
   See here for a quick guide to the grader output.

set-cover.cpp

Choose File No file chosen

Submit