Solutions for CS240 Spring 2023 Midterm Exam

1a. f3 < f1 < f4 < f2

1b. $n^2(\log n)^2$

1c. n^3

1d. true

1e. false. f and g can oscillate

2. We basically do a binary search. Given a list of n Boolean values, let L=OR(1, n/2), R=OR(n/2+1, n). If L=R=F, then there are no T values in the list, and we return F. Otherwise, we recurse on whichever of L or R equals T. When n=1, then we have found a T value.

3. Let M(i) = the max value we can obtain by picking a subset of objects i to n. We have $M(i) = \max(v_i + M(i + f_i + 1), M(i + 1))$, because if we choose item i, then we get v_i plus the max value we can get after skipping f_i items after item i, and if we don't choose item i, then we get the max value we can get from items i+1 to n.

We compute M(n), M(n-1), ..., M(1), where M(1) is the final answer we want. Computing each M(i) takes O(1) time, so the total time and memory complexity are both O(n).

4. Let the sum of all the numbers be X. Focus on the subset with the smaller weight. Then our goal is to find a subset of S whose sum is as close as possible to X/2, without exceeding it. To do this, we use an algorithm similar to subset sum. In particular, we make a table T with n rows and X/2 columns, where T(i,v) is a boolean value indicating whether we can find a subset of values $x_1, \dots, x_i$ which adds up to v. Then $T(i, v) = T(i - 1, v) \vee T(i - 1, v - x_i)$, because we can find a subset of $x_1, \dots, x_i$ adding up to v by either not including x_i and then taking a subset of $x_1, \dots, x_{i-1}$ which adds up to v, or taking x_i and then taking a subset of $x_1, \dots, x_{i-1}$ which adds up to $v - x_i$.

After filling out this table, we look for the max value v such that T(n,v) = 1. Then our output is $X - 2v$, because we have one subset which adds up to v and another which adds up to X-v.

5. There are multiple solutions. The first is an O(n) solution. Find the median of the input list in O(n) time, and use it to partition the input into two sublists, consisting of the values <= the median, and the values > the median. (There's a corner case in which many values are equal; this is easy but somewhat tedious to take care of. You can ignore this case for the exam). Let the two sublists be $x_1, \dots, x_k$ and $y_1, \dots, y_k$. Then output the list $x_1, y_1, x_2, x_2, \dots, x_k, y_k$.

Another solution is to sort the input list into a list $x_1, \dots, x_n$, and output the list $x_1, x_3, x_2, x_4, \dots$ This method takes O(n log n) time.