

Algorithm Theory - Assignment 3

Téo Bouvard

February 22, 2020

Problem 1

a) If we can prove that E has majority $e_m \implies E - \{e_i, e_j\}$ also has e_m as majority, we can devise a simple algorithm that just removes pairs of different elements from the original sequence. At some point, one of two things is going to happen.

- We can't pick a pair of different elements because all elements in the sequence are identical. This implies that this repeating element is the majority of E .
- We have an empty sequence. This implies that E has no majority.

Here is the algorithm in pseudo code, which returns TRUE if E has a majority and FALSE otherwise.

```
MAJORITY( $E$ )
  if  $E = \emptyset$ 
    return FALSE
  if ALL-EQUAL( $E$ )
    return TRUE
  pair = FIND-PAIR( $E$ )
  return MAJORITY( $E \setminus \text{pair}$ )
```

The above algorithm uses two subroutines.

- ALL-EQUAL returns TRUE if all elements of a sequence are equal, and FALSE otherwise. We do not check for empty sequences as this subroutine is called after this check in the MAJORITY algorithm.
- FIND-PAIR returns a sequence of two distinct elements from the input sequence. We only need to consider sequences who actually have at least two distinct elements, because we checked for empty sets and sets containing identical elements before calling this subroutine.

```
ALL-EQUAL( $E$ )
  elem =  $E[0]$ 
  for  $i = 1$  to  $E.length$ 
    if elem  $\neq E[i]$ 
      return FALSE
  return TRUE
```

```
FIND-PAIR( $E$ )
  first =  $E[0]$ 
  for  $k = 1$  to  $E.length$ 
    if first  $\neq E[k]$ 
      return {first,  $E[k]$ }
```

b) This problem has optimal substructure because the optimal solution is constructed from optimal solutions of its subproblems. However, there is no overlap among the subproblems. Dynamic programming requires both optimal substructure and overlapping subproblems to be applied, so we can't use it in this case.

Problem 2

This problem is an instance of the fractional knapsack problem. As Maria can get partial points if a problem is solved partially, her best strategy is to solve the problems with highest point density first. That is, solve the problem with the highest $\frac{p_i}{t_i}$, until either the exam time runs out, in which case the algorithm exits, or the problem is solved, in which case she picks the next problem.

Problem 3