UNIVERSITY COMPUTER SCIENCE • EXERCISE SHEET

10.3 Greedy algorithms

Name: _____ Class: _____ Date: _____ **Total: 24 marks**

KEY WORDS greedy algorithm • exchange argument

Objective

Run greedy algorithms on fixed inputs and prove or refute them by exchange.

1 Key knowledge and worked examples

■ The proof tool

The exchange argument: swap any optimal pick for the greedy pick and stay feasible, same size – repeat, and optimal becomes greedy ✓.

■ Worked example: the counterexample

Coins 1, 3, 4, target 6: greedy pays 4+1+1 (3 coins); the DP optimum is 3+3 (2 coins) – no exchange argument exists ✓.

2 Practice

2.1 Earliest-finish on the course activities picks how many? [3]

2.2 Huffman merges which two nodes first for frequencies 5, 9, 12, 13, 16, 45? [3]

2.3 State the exchange argument in one sentence. [3]

3 Exam-style questions

3.1 Course activities: (1,4), (3,5), (0,6), (5,7), (3,9), (5,9), (6,10), (8,11), (8,12), (2,14), (12,16).

Run earliest-finish-first: list the picks in order with the reject-why for the first two rejections. Prove the first exchange step (swap the optimal's first activity for (1,4)). State the size of the greedy and (from the proof) the optimal schedule. [8]

3.2 Huffman on frequencies 5, 9, 12, 13, 16, 45.

Trace the merge sequence (five merges, smallest two each time). Give the weighted depth (224, suite-verified) and the fixed-length alternative (300 bits). State the leaf-swap argument for optimality in two lines. [7]

Solutions

2.1

- Four – (1,4), (5,7), (8,11), (12,16) ✓.

2.2

- 5 and 9 (merge to 14) ✓.

2.3

- Any optimal solution can swap its picks for greedy's, one at a time, staying feasible and the same size ✓.

3.1

- Picks: (1,4), then (5,7), (8,11), (12,16) ✓.
- First rejections: (3,5) overlaps (1,4); (0,6) overlaps (1,4) ✓.
- Exchange: if the optimal starts with activity $O_1 \neq (1, 4)$, then finish(1,4) $\leq$ finish(O_1), so replacing O_1 by (1,4) leaves every later activity feasible ✓.
- Repeat down the list: the optimal schedule transforms into the greedy one, unchanged in size ✓.
- Greedy size 4 = optimal size 4 ✓.

3.2

- Merges: 5+9 = 14; 12+13 = 25; 14+16 = 30; 25+30 = 55; 45+55 = 100 ✓.
- Weighted depth: $\sum f_i d_i = 224$ bits ✓.
- Fixed length: $\lceil \log_2 6 \rceil = 3$ bits $\times 100 = 300$ ✓.
- Leaf-swap argument: in an optimal tree, the two smallest frequencies sit deepest; if not, swapping them with deeper leaves cannot increase the cost ✓.
- So the greedy merge (two smallest first) is exactly what optimality permits – induction on the merge count closes it ✓.

exercises.lol exercises.com.cn