Unit 1 · Algorithms for Hard Problems

WORKSHEET

Full marks 32

Name _____ Class _____ Roll no. ____ Date _____ Marks _____ / 32

1 Key words [4]

Fill in the blanks with words from the box. Two words in the box are not needed. Each blank carries 0.5 mark.

breadth-first search	**dynamic programming**	**edit distance**	**factorial**	**greedy**	**heuristic**
memoization	**priority queue**	**relaxed**	**weighted**		

A graph with a number on every edge, such as a road's length, is a _____ graph. Dijkstra's algorithm keeps the places it has reached in a _____, so it can always take the nearest. When a shorter way to a waiting place turns up and its distance goes down, the road is said to be _____. A* adds an estimate of the distance left, a rule of thumb called a _____. An algorithm that takes the best-looking step now and never goes back is _____. Remembering a function's answers so that it never works the same one out twice is _____, and solving overlapping smaller problems once each is _____. The fewest inserts, deletes and changes that turn one word into another is their _____.

2 Multiple choice [5]

Choose the correct option for each of the following questions.

i. In Dijkstra's algorithm, which town is visited next? [1]

a. the one that was reached first
b. the one with the fewest roads from the start
c. the waiting town with the smallest distance found so far
d. the waiting town nearest to the goal on the map

ii. An A* search uses an estimate that is sometimes **higher** than the real distance left. What can go wrong? [1]

a. It visits more towns than Dijkstra's algorithm would.
b. It never reaches the goal.
c. It may stop with a route that is not the shortest.
d. Nothing: any estimate gives the shortest route.

iii. Coins are 1, 6 and 10. What does the greedy method give for 12, and is it the fewest? [1]

a. 6 + 6: yes, the fewest
b. 10 + 1 + 1: three coins, but 6 + 6 is only two
c. 10 + 2: two coins
d. 1 × 12: twelve coins

iv. Which problem is a good fit for dynamic programming? **[1]**

a. finding the largest number in a list

b. a problem whose answer is built from answers to smaller problems, where the same smaller problems are needed again and again

c. any problem that can be solved by a loop

d. a problem where the greedy choice is always right

v. A program tries every order of 12 places to find the shortest round trip, and takes 1 second. Roughly how long would 14 places take? **[1]**

a. 2 seconds **b.** 4 seconds

c. about 3 minutes **d.** about a year

3 Dijkstra's algorithm by hand **[4]**

A health worker's trails between six villages, in km (made-up lengths): Arughat–Bhanjyang 4, Arughat–Chisapani 2, Chisapani–Bhanjyang 1, Bhanjyang–Dhunche 5, Chisapani–Dhunche 8, Chisapani–Ekrani 10, Dhunche–Ekrani 2, Dhunche–Fikkal 6, Ekrani–Fikkal 3.

Find the shortest way from Arughat (A) to Fikkal (F) with Dijkstra's algorithm. Write a row for each village visited, in the order visited, with the shortest distance found so far for every village after that visit (– if not reached yet). Write each village by its first letter.

Visited	A	B	C	D	E	F

i. Write the shortest route and its length.

4 Edit distance **[3]**

Fill in the edit-distance table for "bhat" (down) and "bath" (across). Each cell is the smallest of: the cell above + 1, the cell to the left + 1, and the cell above-left + 0 if the two letters are the same or + 1 if not. Then answer.

bhat ↓	–	b	a	t	h
–	0	1	2	3	4
b	1				
h	2				
a	3				
t	4				

i. What is the edit distance, and why does a spell-checker using it find "bhat" no closer to "bath" than to a word two changes away?

5 Fewest coins, by a table [4]

Coins are 1, 4 and 5. Fill in the table upwards, as dynamic programming does: for each amount, the fewest coins that make it, and the last coin of that best way. Then answer.

Amount a	best[a]	Last coin
0	0	–

i. What is the fewest number of coins for 8, which coins are they, and how many does the greedy method (the biggest coin first) use?

6 What does it print? [3]

Write exactly what the program prints, in the box under it.

```
1  def greedy_change(amount, coins):
2      given = []
3      for coin in sorted(coins, reverse=True):
4          while amount >= coin:
5              given.append(coin)
6              amount = amount - coin
7      return given
8
9  print(greedy_change(8, [1, 4, 5]))
10 print(greedy_change(13, [5, 1, 4]))
11 print(len(greedy_change(785, [1, 2, 5, 10, 20, 50, 100, 500, 1000])))
```

OUTPUT

7 Find and fix the bug [3]

Read the program, then answer.

The program should find the shortest distance on the trails of question 3, each village written by its first letter. It should print 13, but it prints 14.

```
import heapq

trails = {"A": {"B": 4, "C": 2}, "B": {"A": 4, "C": 1, "D": 5},
          "C": {"A": 2, "B": 1, "D": 8, "E": 10},
          "D": {"B": 5, "C": 8, "E": 2, "F": 6},
          "E": {"C": 10, "D": 2, "F": 3}, "F": {"D": 6, "E": 3}}

def shortest(graph, start, goal):
    best = {start: 0}
    done = set()
    waiting = [(0, start)]
    while len(waiting) > 0:
        km, place = heapq.heappop(waiting)
        if place in done:
            continue
        done.add(place)
        for nxt, trail_km in graph[place].items():
            new_km = km + trail_km
            if nxt == goal:
                return new_km
            if nxt not in best or new_km < best[nxt]:
                best[nxt] = new_km
                heapq.heappush(waiting, (new_km, nxt))
    return None

print(shortest(trails, "A", "F"))
```

i. Why does it give 14?

ii. Write the lines that fix it, and say where they go.

8 Think [2]

Think, then answer.

Mina's route app uses A* with an estimate of 1.5 times the straight-line distance. She says: "It visits fewer places than with the plain straight line, so it is faster, and it still always finds the shortest route." Is she right? Explain.

9 Write a program [4]

Write the program in Python on the lines. Any program that gives the right result gets the marks.

The school hall can be booked for one event at a time. Write a function `book_hall(requests)` that takes a list of requests, each a pair `(start, end)` in hours, and gives back the most events that can fit, chosen by the greedy rule of the unit: take the request that finishes earliest, then, each time, the one that finishes earliest among those starting at or after the last one booked ends. Then use it on `[(9, 11), (10, 12), (11, 13), (8, 10), (12, 14), (13, 16), (14, 15)]`.

10 Activity: Plan the trek (not marked)

At a computer or on a phone, open the unit's page (the QR code) and find **Plan the trek.** On the map of 9 villages, tap your own order first, then try each method. Fill in the table.

Method	Length of the trek (km)	How much longer than the best (%)
My own order		
Nearest next		
Nearest next, then uncross		
Try every route (the best)		0