

Problem 3.3

Suppose two friends live in different cities on a map, such as the Romania map shown in Figure 3.2. On every turn, we can simultaneously move each friend to a neighboring city on the map. The amount of time needed to move from city i to neighbor j is equal to the road distance $d(i, j)$ between the cities, but on each turn the friend that arrives first must wait until the other one arrives (and calls the first on his/her cell phone) before the next turn can begin. We want the two friends to meet as quickly as possible.

- a. Write a detailed formulation for this search problem. (You will find it helpful to define some formal notation here.) b. Let $D(i, j)$ be the straight-line distance between cities i and j . Which of the following heuristic functions are admissible? (i) $D(i, j)$;
(ii) $2 \cdot D(i, j)$;
(iii) $D(i, j)/2$.
c. Are there completely connected maps for which no solution exists? d. Are there maps in which all solutions require one friend to visit the same city twice?

Part A

This search problem consists of the following components: **Goal:** place both people, i.e. p_1 and p_2 in the same city i , in the lowest possible time T .

Define $\text{dist}(i, j)$ as the straight-line distance between two cities i and j .

Problem:

- States: the pair of cities i, j , that p_1 and p_2 , respectively, are currently located in. i and j may represent the same city.
- Actions: moving p_1 to city i' (i -prime) and p_2 to city j' (j -prime), incurring an addition to T of $\max(\text{dist}(i, i'), \text{dist}(j, j'))$

Solution:

The solution is two equal-length sequences of cities, one for p_1 and one for p_2 , in the form $i \rightarrow j \rightarrow \dots \rightarrow k \rightarrow l \rightarrow m$ and $a \rightarrow b \rightarrow \dots \rightarrow c \rightarrow d \rightarrow m$. The final city must be in each sequence, and the sequences must not share any other cities *in the same index of the sequence*. Additionally, the solution must minimize the following function:

```
// both of length n
ds1 = distances traveled by p1 in sequence 1
ds2 = distances traveled by p2 in sequence 2

// get the largest distance for each position
dist = map(max, zip(ds1, ds2))
```

This results in the two people being in the *same place* with the shortest overall distance traveled.

Part B

The heuristic $D(i, j)/2$ is admissible. In the ideal case, both people proceed toward each other at equal pace, resulting in twice the distance reduction compared to if one of them had simply walked to the other. Therefore, the solution becomes more optimal as it approaches the heuristic.

Part C

There are completely connected maps for which no solution exists. The easiest example is a two node graph, i.e.

(P1)------(P2)

As both have to move at once, they will simply swap places and can never be in the same city:

(P1)------(P2)

|

v

(P2)------(P1)

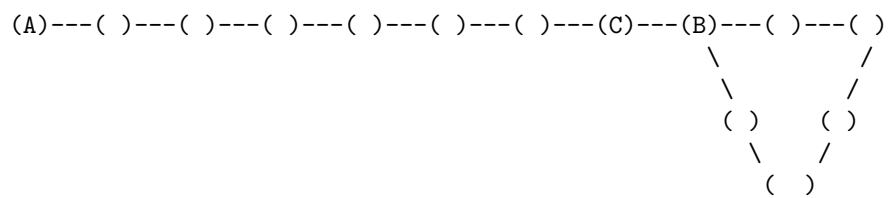
|

v

(P1)------(P2)

Part D

Yes, in specific cases all solutions require a person to visit the same node twice, e.g.



A and B above can meet at C, but this requires B to go around the loop and return to B. All other solutions (with more steps) would require either backtracking or for A to move around the loop and return to its entry point of it as well.