Aleksandr Madisson

Homework 5

Question 1

The number of executions for the line 8 depends on the dictionary D.
If D is emtpy then A.len - 1 times, if D contains all computed routes
then 0. If init state for D is D.len = 0, then for the first Compute-Route(A, D)
call D would be called for A.len - 1 times, for the second time D would be
called for 0 times, because D must already contain best routes.

Question 2

A new k_A,D value will be increased after adding a new location, because in
this case the D will contain pairs which don't have computed paths.
No paths for between A[i-1], A[i] and A[i], A[i+1].

I guess these values depend on how many times the COMPUTE-ROUTE was called
before comparing k_A',D with k_A,D.
The difference range must be in 0 <= k_A',D - k_A,D <= 2.

Questions 3

Operation	Actual cost c_i	Amortized cost c_i
Compute-Route(A, D)	1 + (k_A,D v)	1
Insert(A, i, a)	1	2v + 1

Very first call of Compute-Route(A, D) where init A contains a few locations
and D is empty will be a most expensive, for this call we could allocate a credit
or a load. Insert operation is lightweight and can return a credit. Next
Compute-Route(A, D) are less expensive because will contain computed paths.

We already know that k_A,D <= 2, then 2v + 1 + 1 => 1 + (k_A,D v) + 1, then
2v => k_A,Dv

Question 4

It's correct because there are no incorrectness, and because $\mathbf{k_i}$ depends on
D which gets changed, but the route computation has a specific complexity **v**. In order to
leverage a complexity of the Compute-Route(A, D) we can play around with A or/and D but the
algorithm itself **v** won't be change.

Cost for each operation, $c_i' = c_i + \Phi(D_i) - \Phi(D_{i-1}) =$
$c_i + k_i v - k_{i-1} v$.

Question 5

If this exercise is supposed to be solved without k_A,D ("In terms of n and v"), then:
Amortization cost T(n) = [(1 + v) + 1 + (1 + v) + 1 + (1 + v) + 1 + ...] / n =
[1 + 1 + 1 ... + (1 + v) + (1 + v) + (1 + v) ...] / n =
[n + n(1 + v)] / n = [n + n + nv] / n = [2n + nv] / n = 2 + v

Date: 12.04.2021