On question 11 and 12 under Linear sorting fundamentals, what does the question mean by comparing "entire" elements with one another? Is it asking if the algorithm compares the object (not just the object's key) against the other objects?

Quote: "T or F:
A linear-time sort does not compare entire elements with one another."

I mean the entire key.

Can someone explain to me how to find the Potential Function for these questions? I remember (amortized cost = actual cost + \phi(i) - \phi(i-1)) but I do not know how to apply this here. I have no idea why the smiley faces appeared in my question.

Question 72, 73, and 74

Also, for question number 2. What is the full distinction between "case" and "situation". If the input list to Bubble Sort is already sorted, would this be best case or best situation? I would think that would be situation.

In question 72,
for A,
actual cost = 1
and its given after an operation n increase by 1. so, n+1

Also, for B,
actual cost = 2n+1
and after an operation n decrease to 0.
we can use hit and trail method using the options.

lets consider option a) \( \phi = n \)

In case of A,
we know,

amortized cost = actual cost + \( \phi_D(i) - \phi_D(i-1) \)

\( \phi_D_A(i) = \) current state of A
\( \phi_D_A(i-1) = \) previous state of A

So,

amortized cost_A = actual cost_A + \( \phi_D_A(i) - \phi_D_A(i-1) \)

= 1 + (n+1) - n  Since, after an operation n increase by 1,
= 2 so if before the operation the state of A was n

= constant (\( \phi_D_A(i-1) = n \)) then \( \phi_D_A(i) = n+1 \) (current state)
= shows amortized bound of theta(1)

In case of B,

\( \phi_D_B(i) = \) current state of B = 0
\( \phi_D_B(i-1) = \) previous state of B = n

amortized cost_B = actual cost_B + \( \phi_D_B(i) - \phi_D_B(i-1) \)

= 2n+1 + 0 - n
= n+1
= linear
= does not show amortized bound of theta(1)

Since both A & B should show amortized bound of theta(1) \( \phi = n \) is not correct.

now,

lets consider option b) \( \phi = 2n \)

In case of A,

amortized cost_A = actual cost_A + \( \phi_D_A(i) - \phi_D_A(i-1) \)

= 1 + 2*(n+1) - 2*n
= 1 + 2n + 2 - 2n
= 3 (constant)
= shows amortized bound of theta(1)

In case of B,

remember \( \phi = 2n \)

amortized cost_B = actual cost_B + \( \phi_D_B(i) - \phi_D_B(i-1) \)

= 2n+1 + 2*0 - 2n
= 2n + 1 - 2n
= 1 (constant)
= shows amortized bound of theta(1)

Therefore, the solution is \( \phi = 2n \).

you can imply same principle to 73 and 74

73 is mistake. In 73, real cost for A should be 3n+1 and real cost of B should be (3/2)n + 1
Wood McGowan wrote on Wed, 27 January 2016 19:26 I have no idea why the smiley faces appeared in my question.

When posting, click on 'Disable smilies in this message' under the options menu, which is under the area where you type your message.

Questions in "The classes of NP complete problems" #'s 102 - 105 ask if NP problems solvable in x time are NP or P. Are they either as long as they are O-polynomial)?