10 Algorithms 2 (djw1005)

Consider a stack which supports `push` and `pop`, which we would like to back up periodically by copying it to disk. We can do this by keeping a counter x, initialized to 0 and incremented by 1 on every `push` or `pop`; if this increment results in $x = N$ then we copy the current stack to disk and reset x to 0. The cost of copying the stack to disk is proportional to the number of items on the stack, and the cost of pushing or popping an item (but not including the cost of copying to disk) is $O(1)$.

(*a*) Define *potential function.* Given a potential function, explain how one can use it to obtain amortized costs. [3 marks]

(*b*) Suppose that the stack size never exceeds N items. Using a potential function, or otherwise, show that the amortized costs of both `push` and `pop` are $O(1)$, asymptotic in N. [8 marks]

(*c*) Suppose there is no restriction on stack size. Show that the amortized costs cannot both be $O(1)$. [9 marks]