

Due at the start of class Monday, June 7, 2010.

Problem 1. Assume you have a necklace of stones. Some of the stones have positive value and some have negative value. You have the opportunity to snip the necklace in two places (creating two bands) and weld the endpoints of one of the two bands back into a necklace. You would like your new necklace to be as valuable as possible.

(a) Give an algorithm to find the value of the new necklace. You can assume the necklace has n stones with values $v[0], v[1], \dots, v[n-1]$. If all of the stones have negative value your answer should be 0. Make your algorithm as clean and elegant as possible.

(b) Give an algorithm to determine where you should snip the original necklace (not just its value). Make your algorithm as clean and elegant as possible. If all of the values are positive you should not snip and your algorithm should print:

Do not snip.

If all of the values are negative you should not snip and your algorithm should print:

Throw necklace away.

If possible the algorithm should determine these two situations without explicitly checking for them.

Problem 2. Assume you take a sorted list and randomly pick out an element and put it back into a random location. Assume you execute insertion sort with a sentinel on this list.

(a) What is the worst case number of comparisons? Justify.

(b) What is the worst case number of moves? Justify.

(c) What is the average case number of comparisons? Justify.

(d) What is the average case number of moves? Justify.