1. No, for example, the optimal multiple alignment of CCCCTTTT, TTTTGGGG, and GGGGCCCC does not produce any of the optimal pairwise alignments.
2. Optimal alignment is:

ACTACA-

---ACAA

It has a score of -1

1. Instead of finding a node on the midpoint, we should search for a node on the diagonal from the bottom left to the top right, as this will ensure that the two partitions more equally balanced in size.
2. (+1 -3 -4 -2 +6 +7 -5 +8)

(+1 +2 +4 +3 +6 +7 -5 +8)

(+1 +2 -3 -4 +6 +7 -5 +8)

(+1 +2 +3 -4 +6 +7 -5 +8)

(+1 +2 +3 +4 +6 +7 -5 +8)

(+1 +2 +3 +4 +5 -7 -6 +8)

(+1 +2 +3 +4 +5 +6 +7 +8)

(+1|-3|-4|-2|+6 +7|-5|+8)

There are 3 cycles.

Minimum number of reversals is 5.

1. Yes, because the probability of a breakage in a fragile region is linear with respect to the size of the region, which, when the fragile regions are randomly distributed, is the same as randomly selecting breakpoints in the entire genome.
2. Translocating (1 -2 3 4) and (-5 -6 -7 8) into two linear chromosomes (1 -2 -7 8) and (-5 -6 3 4)
3. First, find all kmers of the desired length, and sort. Then, traverse through each list with a pointer. If both are the same, add to the result. If not, advance the pointer pointing to the one that is lesser in lexicographic order. The runtime will be O(m + n). The sorting will take O(k) time, since we can use a radix sort.
4. No. Assume the graph has the following structure.

Thus, we can see that:

km = (ik + jk – ij)/2

= (4 + 4 – 3)/2

= 2.5

im = 4 – 2.5

= 1.5

jm = 3 – 1.5

= 1.5

in = (ik + il – kl)/2

= (4 + 3 – 2)/2

= 2.5

kn = 4 – 2.5

= 1.5

ln = 2 – 1.5

= 0.5

mn = ik – im – kn

= 4 – 1.5 – 1.5

= 1

jm + mn + ln = 1.5 + 1 + 0.5

= 3

!= jl

Thus, not additive.