P4

Prompt:

Given n real numbers, where n is even, design a polynomial time algorithm to partition these numbers into $\frac{n}{2}$ pairs that minimizes the sum of the maximim partition.

Algorithm:

```
// Where R is a set of n real numbers and n is even
Partition(R):
    Sort R
    Initialize S to be an empty list
    While R not empty
        Add {r_0, r_n} to S
        Remove r_0 and r_n from R
```

Claim:

The algorithm terminates in polynomial time.

Proof:

We see that the algorithm terminates because every iteration of the while loop removes two terms from R, thus R will eventually be empty. Depending on the sort algorithm chosen for this algorithm, the runtime of this step could range from O(1) to O(n^2). We'll assume the latter for all inputs of R. As we've already noted, the while loop removes two terms from R on every iteration for a runtime of O(n). Thus because the runtime we've assumed for the sort algorithm is the dominant factor, on all inputs the algorithm runs in O(n^2) which is polynomial. $\square$

Claim:

The algorithm finds the optimal partitioning of n numbers, where n is even, into $\frac{n}{2}$ pairs that minimizes the sum of the maximim partition.

Proof:

Let K be the given sequence. Sort K and number it's terms such that K = $\{k_0$, ..., $k_{\frac{n}{2}}$, $k_{\frac{n}{2}+1}$, ..., $k_n\}$. The algorithm forms pairs $\{k_0, k_n\}$, $\{k_1, k_{n-1}\}$, $\{k_2, k_{n-2}\}$ such that the first pair consists of the outermost values, the second pair of values each one term in from the outside, and so on. Let the optimal pair found by the algorithm be $\{k_i, k_j\}$ such that $k_0 \leq k_i \leq k_{\frac{n}{2}}$ and $k_{\frac{n}{2}+1} \leq k_j \leq k_n$. Now, assume for a contradiction that the pair $\{k_i, k_j\}$ returned by the algorithm is not optimal. Notice that k_i cannot pair with any k_p such that $k_j < k_p <= k_n$ because k_p is already paired with k_q such that $k_0 \leq k_q < k_i$. For the same reason, k_j cannot pair with any k_s such that $k_i < k_s \leq k_{\frac{n}{2}}$. Thus, k_i or k_j pairing with partners in those ranges would create a pair larger than the previous maximum which would be a contradiction. Therefore, there must exist some $k_{\frac{n}{2}+1} \leq k_n < k_j$ and $k_0 \leq k_m < k_i$ for new pairs $\{k_i,\ k_n\}$ and $\{k_m,\ k_j\}$. But because k_m and k_n had partners they left for k_i and k_j, those partners need new partners. By the same reasoning from above, these partners cannot pair with partners of higher value so they too must form pairs with lower valued partners. And this continues with each new pair formed, leaving previously paired parners needing new partners. Until $k_{\frac{n}{2}}$ or k_n need new partners. As we have already shown, neither can have a higher valued partner than they already had. Neither can they have a lower valued partner than they already had because there are no lower valued partners left. This is a contradiction. Thus the original pairing was optimal. $\square$