Problem 1, Defensive Backs Assignment

Solution. The proposed algorithm goes as follows:

```
Sort R[] the wide receivers by descending heights
Sort B[] the defensive backs by descending heights

Start with empty assignment {}

for i = 1, ..., n do:
  assign B[i] to defend R[i]
return assignment
```

It is trivial that this algorithm indeed outputs a matching M. We claim M is optimal. To see this, let $M' \subset D[] \times R[]$ be any optimal matching and suppose it is different from M. By the exchange argument shown in lecture, some i^{th} and $(i+1)^{\text{th}}$ entries of M' is out of order with respect to M, and it suffices to show that swapping *such* adjacent out-of-order indices preserves optimality. For notational convenience we suppose M assigns height b_i to defend r_i and b_{i+1} to defend r_{r+1}, where $r_i \geqslant r_{i+1}$ and $b_i \geqslant b_{i+1}$. Hence M' assigns b_{i+1} to defend r_i and b_i to defend r_{i+1}. AM-GM implies that given a fixed product, the sum of the two numbers is smaller when they are closer. In this case the product

$$2^{r_i - b_i} \cdot 2^{r_{i+1} - b_{i+1}} = 2^{r_i - b_{i+1}} \cdot 2^{r_{i+1} - b_i},$$

and by assumption

$$(r_i - b_i) - (r_{i+1} - b_{i+1}) = r_i - r_{i+1} - (b_i - b_{i+1}) \leqslant r_i - r_{i+1} - (b_{i+1} - b_i) = (r_i - b_{i+1}) - (r_{i+1} - b_i).$$

That is,

$$2^{r_i - b_i} + 2^{r_{i+1} - b_{i+1}} \leqslant 2^{r_i - b_{i+1}} + 2^{r_{i+1} - b_i},$$

so swapping the out-of-order indices of M' does not increase expected gain. After each adjacency swap, the number of out-of-order pairs reduces by 1, so after finite operations we recover M, and in doing so we have proven M is optimal.

Runtime: sorting the two data sets both take $\mathcal{O}(n \log n)$; making assignment takes n iterations with $\mathcal{O}(1)$ time each. Therefore the total runtime is $\mathcal{O}(n \log n)$ which is indeed asymptotically bounded by n^2.

Problem 2, Weight Selection

Proof of 2(a). Let T be given and let $R(T)$ be the collection of weights representable by sums of subsets of T. It is immediately clear that $|R(T)| \leqslant |\mathcal{P}(T) \backslash \varnothing| = 2^n - 1$ (where $\mathcal{P}$ denotes the power set), since the mapping

$$\psi : \mathcal{P}(T) \backslash \varnothing \to R(T) \qquad \text{defined by} \qquad \psi(A) := \sum_{k \in A} k \tag{*}$$

is by construction a surjection.

On the other hand, we note that $S = \{2^j\}_{0 \leqslant j \leqslant n-1}$. To see this, we induct on the following claim:

> After the m^{th} iteration, $S = \{2^j\}_{0 \leqslant j \leqslant m-1}$ and $R(S) = \{1, \ldots, 2^m - 1\}$.

The base case $n = 1$ is trivial. Assuming the claim holds for k, see that 2^k is the smallest integer not representable by S so we add it to S. Since we can originally represent every integer in $[1, 2^k - 1]$, by adding 2^k to it, we can now represent any number in $[2^k + 1, 2^{k+1} - 1]$ as well. And of course 2^k itself is representable by $\{2^k\}$. Hence the $(k+1)^{\text{th}}$ iteration results in $S = \{2^j\}_{0 \leqslant j \leqslant k}$ with $R(S) = \{1, \ldots, 2^{k+1} - 1\}$, completing the induction.

After n iterations, $R(S)$ precisely contains $2^n - 1$ consecutive integers. This in conjunction with (*) proves the optimality of S. □

Proof of 2(b). Done above. □