1. From the list of functions given in class, \( (\log n)^2 = o(n^{0.5}) \). Thus, \( 5n^{2.5} + 100n^2(\log n)^2 + 10^{1000} = \Theta(n^{2.5}) \). The given statement is thus true.

   \( 2^{2n \log n} = (2^{\log n})^{2n} = n^{2n} = (n^n)^2 \). Thus, \( n^n = O(2^{2n \log n}) \) but \( n^n \neq \Omega(2^{2n \log n}) \). The given statement is false.

2. Run time of Test is \( \sum_{i=1}^{3n} \sum_{j=1}^{5^{i^2}+4i} 1 = \sum_{i=1}^{3n} (5^{i^2}+4i) = 5 \frac{3n(3n+1)(6n+1)}{6} + 4 \frac{3n(3n+1)}{2} = \Theta(n^3) \).

3. Consider the following algorithm:

   \( \text{for } i := 1 \text{ to } \alpha n^{2/3} \log_e n \text{ do} \)

   \( \quad \text{Pick a random } j \in [1, n] \text{ and pick a random } k \in [1, n]; \text{ If } j \neq k \text{ and } a[j] = a[k] \)

   \( \quad \text{then output: "Type II" and quit; } \)

   Output: "Type I";

   **Analysis:** Note that if the array is of type I, the above algorithm will never give an incorrect answer. Thus assume that the array is of type II. We’ll calculate the probability of an incorrect answer as follows.

   Probability of coming up with the correct answer in one iteration of the for loop is

   \( \frac{n^{2/3}(n^{2/3}-1)}{n^2} \approx \frac{1}{n^{2/3}} \). Thus, probability of failure in any iteration is \( 1 - \frac{1}{n^{2/3}} \). As a consequence, probability of failure in \( q \) successive iterations is \( (1 - \frac{1}{n^{2/3}})^q \leq \exp(-q/n^{2/3}) \) (using the fact that \((1 - 1/x)^x \leq 1/e \) for any \( x > 0 \)). This probability will be \( \leq n^{-\alpha} \) when \( q \geq \alpha n^{2/3} \log_e n \).

   Thus the output of this algorithm is correct with high probability.

4. Keep two 2-3 trees \( N \) and \( S \). In \( N \) store all the records with the name as the key for each record and in \( S \) store all the records with the social security number as the key for each record. To process \( \text{Find Name}(SSN) \), we search for a record whose key is \( SSN \) in the tree \( S \). The name in this record will be output. The run time is \( O(\log n) \). We process \( \text{Find SSN}(Name) \) in a similar manner.

5. There are five calls made to \( \text{Heapify} \). The tree takes the following shape after these five calls:

   \( 23, 12, 5, 6, 34, 17, 14, 8, 2, 11, 21; \quad 23, 12, 5, 8, 34, 17, 14, 6, 2, 11, 21; \quad 23, 12, 17, 8, 34, 5, 14, 6, 2, 11, 21; \quad 23, 34, 17, 8, 21, 5, 14, 6, 2, 11, 12; \quad 34, 23, 17, 8, 21, 5, 14, 6, 2, 11, 12. \)
6. Recurrence relation for the run time of \( \mathcal{A} \) is: 
\[
T(n) = 16T(n/2) + n^3.
\]
Here \( a = 16, b = 2, f(n) = n^3, n^{\log_b a} = n^4 \). Case 1 of Master theorem applies. Thus, \( T(n) = \Theta(n^4) \).

Recurrence relation for the run time of \( \mathcal{B} \) is: 
\[
T(n) = 64T(n/8) + n^2.
\]
Here \( a = 64, b = 8, f(n) = n^2, n^{\log_b a} = n^2 \). Case 2 of Master theorem applies implying that 
\( T(n) = \Theta(n^2 \log n) \).

Therefore, algorithm \( \mathcal{B} \) is preferable.