Midterm Solutions

Problem 1:

Set $T(n)$ to be the number of instructions executed for an array of length *n*. Assume $n > 10$. Then reading of the code, we get

$$T(n) = 6T((3/5)n)+\text{const}$$

By making the constant large enough, we can assume that this formula is true for all $n > 0$. For the master theorem, we have $a = 6$ and $b = 5/3$. Thus, we need to compare n^c, with $c = \log_{5/3}(6) \approx 3.5$, with the constant function const. Clearly, const $\in O(n^{c-0.5})$, so that $T(n) = \Theta(n^c)$.

Problem 2:

We go through the array of characters and count the number of occurrences of each. This takes proportional to `n=len(array)` steps. Assume that the counts are c_b, c_w, and c_r respectively. We then print out c_b characters 'b', c_w, characters 'c', and c_r characters 'r'. This also takes proportional to `n=len(array)` steps. In total, this counting based algorithm is linear.

Problem 3:

We process the array from left to right. We initialize MSCS and MLR to the first element. We then look at the next element. MLR is either the current element or the MLR plus the current element, We select the larger one. Then we update MSCS to be the maximum of the current MSCS and the updated MLR.

```
def max_cont_sum(array):
    mlr = array[0]
    mscs = array[0]
    for element in array[1:]:
        if mlr >= 0:
            mlr = mlr+element
        else:
            mlr = element
        if mscs < mlr:
            mscs = mlr
    return mscs
```

Problem 4:

Yes. If we have two floating point numbers x and y, then $x \cdot y = \frac{(x+y)^2 - x^2 - y^2}{2}$. A multiplication can be implemented with 3 squaring, two subtractions, and one division by 2.

Problem 5:

Since $19 = 2^4 + 3$, the current level is 4 and the current split pointer is 3. The record with key-hash $33 = 2 \cdot 16 + 1$ has to be recalculated, since $1 < 3$. But $33 = 32 + 1$, so the record goes to Bucket 1. Since $40 = 32 + 8$, this record goes to Bucket 8. Since $49 = 3 \cdot 16 + 1$, and $1 < 3$, we need to recalculate: $49 = 1 * 32 + 17$, so the last record goes to Bucket 17.

Problem 6: