Homework 4 Solutions

1.

```
merge3(A, l, m1, m2, u)
{
     merge(A, l, m1, m2);
     merge(A, l, m2, u);
}


mergesort3(A[l..u])
{
     if (l < u)
     {
             s = (l - u + 1) / 3;
             mergesort3(A[l..l+s-1]);
             mergesort3(A[l+s..l+2*s-1]);
             mergesort3(A[l+2*s..u];
             merge3(A, l , l+s-1, l+2*s -1, u);
     }
}
```

Let $C(n)$ be the worst-case number of comparisons made by `mergesort3` on an array of size n. We have the recurrence:

$$\begin{aligned} C(1) &= 0 \\ C(n) &= C(\lfloor \frac{n}{3} \rfloor) + C(\lfloor \frac{n}{3} \rfloor) + C(n - 2\lfloor \frac{n}{3} \rfloor) + 2(\lfloor \frac{n}{3} \rfloor) - 1 + (n-1), n \geqslant 2. \end{aligned}$$

For $n = 3, 9, 27, \ldots$, the recurrence becomes

$$C(n) = 3C(\frac{n}{3}) + \frac{5}{3}n - 2$$

Applying the Master Theorem, we have $a = b = 3$, $d = 1$, and since $a = b^d$,

$$C(n) \in \Theta(n \lg n).$$

Asymptotically speaking, subdividing the original problems into more than 2 subproblems does not improve the running time.

2. We prove that the greedy algorithm works for the coin change problem when the denominations are $1, c, c^2, \ldots, c^k$ for $c \geqslant 1$ and $k \geqslant 0$.

Proposition 1. *An optimal solution S for n contains a coin $c^{\min(d,k)}$, where $c^d \leqslant n < c^{d+1}$.*

Proof. Let d be the integer such that $c^d \leqslant n < c^{d+1}$. Since S is optimal, S contains at most $c-1$ coins c^e for any $e < \min(d,k)$; furthermore, it cannot contain any coin c^e, $e > \min(d,k)$. The total values of coins of denomination less than $c^{\min(d,k)}$ is at most

$$\begin{aligned} \sum_{e=0}^{\min(d,k)-1} (c-1)c^e &= (c-1)\frac{c^{\min(d,k)} - 1}{c-1} \\ &= c^{\min(d,k)} - 1 < n \end{aligned}$$

so there must be a coin of denomination $c^{\min(d,k)}$ in S. $\square$

It is also obvious that the remaining coins in S must form an optimal solution for $n - c^{\min(d,k)}$. Hence the greedy strategy yields an optimal solution for this problem.