Ex 1. (1) First, to choose a k-key combination from n keys, there are $\binom{n}{k}$ different combinations

For a certain k-key combination, the possibility that they lie into the slot is $\left(\frac{1}{n}\right)^k$. For all the other keys, the possibility that they do not go into that slot is $(1-\frac{1}{n})^{n-k}$

In conclusion, the possibility that k keys go into a slot is

$$(\frac{1}{n})^k(1-\frac{1}{n})^{n-k}\binom{n}{k}$$

(2) From (1) we can see that for any integer between 1 and n, the possibility that the i_th slot has k keys is P_k. However since there is possibility that more than one slot has k keys, the possibility that there exists a slot which has k keys is smaller than nP_k. And the possibility that that slot has most number of keys is even smaller so

$$P_k^{'} <= P_k$$

(3) Because $\binom{n}{k} = \frac{n!}{k!(n-k)!}$

According to stirling formula, we have

$$\binom{n}{k} < \frac{en^{n+\frac{1}{2}}e^{-n}}{2\pi k^{k+\frac{1}{2}}(n-k)^{n-k+\frac{1}{2}}e^{-n}}$$

Then

$$P_k < \frac{e\sqrt{n}(n-1)^{n-k}}{2\pi k^{k+\frac{1}{2}}(n-k)^{n-k+\frac{1}{2}}} < \frac{e\sqrt{n}}{2\pi k^{k+\frac{1}{2}}\sqrt{n-k}} \times e^{k-1} < \frac{e^k}{k^k}$$

Ex 2. The algorithm is given below

Algorithm 1 Finding Minimal Spanning Tree with Decreased Edge Weight

Input: Graph $G =< V, E >$, previous Minimal Spanning Tree T, the updated edge $e_0 = (m, n)$ and its new weight w_e

Output: A new Minimal Spanning Tree for G

```
1: $E_p \leftarrow Edges\ on\ the\ path\ from\ m\ to\ n\ in$
2: $e_t \leftarrow The\ edge\ in\ E_p\ with\ greatest\ weight$
3: if $w_{e_t} > w_e$ then
4:     $T \leftarrow T \setminus e_t$
5:     $T \leftarrow T + e$
6: end if
7: return T
```

Ex 3. (2). (a) The algorithm is shown below.

Algorithm 2 Multiplication using recursion

```
Input: Two numbers $x, y$
Output: Their multiplication
function MULTI($x, y$)
    if $x = 0$ or $y = 0$ then
        return 0
    else
        if $y \equiv 0 (mod\ 2)$ then
            return $Multi(x * 2, y/2)$
        else
            return $Multi(x * 2, (y - 1)/2) + x$
        end if
    end if
end function
return $Multi(x, y)$
```

(b) Consider if y is even then $xy = (2x) \times \frac{y}{2}$

If y is odd then $xy = x(1 + 2 \times \frac{y-1}{2}) = x + (2x) \times \frac{y-1}{2}$

So it can be seen that the recurrance relation is correct. Then because the parameter y is divided by 2 repeatedly, it will finally com to 0, so the boundary condition is reachable.

So the algorithm is correct.

Ex 4. 7 races should be applied to those horses. The algorithm is shown below.

First, devide the 25 horses into 5 groups of five horses, and perform a race for each group.

Then pick all the 5 No. 1 horses and let them do the race. Consider in this race a_i represents the i_th fastest horse, and obviously a_1 is the No.1 horse in all 25 horses.

Then perform the last race for the No.2 and No.3 horse in a_1's original group, a_2 and the No.2 horse in his group and a_3. The fasetest and second fastest horse will be the No.2 and No.3 horse for all 25.

Ex 5. (1) The two algorithms can both solve the problem in some situation, but cannot for some others.

Consider the following two situations:

1. $S = \{2, 4, 5\}, n = 5$
2. $S = \{2, 4, 5\}, n = 6$

It is obvious that the first algorithm can solve the second situation but not the first, while the situation for the second algorithm is just the opposite.

So in conclusion, the two both work in some situation but both have their limits.

(3). The problem is the **coin change problem**.

Description: Giving a method of changing \$n money with $\$a_1, \$a_2, ..., \$a_n$ coins, which uses the smallest number of coins.

Greedy Algorithm: Always use the coin with the largest value to change, until the remaining number is smaller than that value. Then uses the second largest value coins and repeat the whole process.

Counter Example: When $n = 15$, $a = \{1, 5, 6, 7\}$, The greedy algorithm will give the method that uses $1 \times \$7, 1 \times \$6$ and $2 \times \$1$, which costs 4 coins. But in fact the globally optimal answer is using three \$5 coins.