Algorithmn HW3

5140379032 JIN YI FAN

Problem 1.38

```
Require: set of $n$ positive integers $S$
Ensure: two subsets $S_1$, $S_2$ with maximum sum difference
$QuickSort(S[])$
$S_1 \leftarrow S[0 : n/2 + 1]$
$S_2 \leftarrow S[n/2 + 1 : n]$
return $S_1$, $S_2$
```

The time complexity is $O(n \log n)$

Problem 1.39

```
Require: set of $n$ positive integers $S$
Ensure: two subsets $S_1$, $S_2$ with minimum sum difference
function NEXT_PERM($A[]$)
    count the next permutation of elements in $A[]$
    return an array containing this permutation.
end function
$dif \leftarrow Max\_int$
$whole \leftarrow sum(S[])$
while $next_Perm$ exists do
    $S[] \leftarrow$ NEXT_PERM($S[]$)
    $tempDif \leftarrow (whole - 2sum(S[0 : n/2]))$
    if $tempDif < dif$ then
        $dif \leftarrow tempDif$
        $Candidiate \leftarrow S[]$
    end if
end while
$S_1 \leftarrow Candidate[0 : n/2]$
$S_2 \leftarrow Candidate[n/2 + 1 : n]$
return $S_1$, $S_2$
```

The time complexity is $O(n^2 \cdot sum)$

Problem 5.4

```
Require: an array with $n$ real numbers $A[1 \dots n]$
Ensure: the average number
function AVE($A[], k$)
    if $k$ is 1 then
        return $A[0]$
    else
```

```
        $res \leftarrow (\text{AVE}(A[], k-1)*(k-1) + A[k])/k$
        return $res$
    end if
end function
return $\text{AVE}(A[], n)$
```

Problem 5.7

(a) 4567, 2463, 6523, 7461, 4251, 3241, 6491, 7563

Step 1:	**Step 2:**	**Step 3:**	**Step 4:**
1:7461,4251,3241	2:6523	2:3241,4251	2:2463
2:6492	4:3241	4:7461,2463,6492	3:3241
3:2463,6523,7563	5:4251	5:6523,7563,4567	4:4251,4567
7:4567	6:7461,2463,7563,4567		6:6492,6523
	9:6492		7:7461,7563

(b) 16543, 25895, 18674, 98256, 91428, 73234, 16597, 73195

Step 1:	**Step 2:**	**Step 3:**	**Step 4:**	**Step 5:**
3:16543	2:91428	1:73195	1:91428	1:16543,16597,18674
4:18674,73234	3:73234	2:73234,98256	3:73195,73234	2:25895
5:25895,73195	4:16543	4:91428	5:25895	7:73195,73234
6:98256	5:98256	5:16543,16597,18674	6:16543,16597	9:91428,98256
7:16597	8:18674	8:25895	8:98256,18674	
8:91428	9:25895,73195,16597			

Sorting variable-length items

Require: $L[]$ containing n ints in $range[0 : n^3 - 1]$

Ensure: Sorted array within $O(n)$ time

```
$k \leftarrow \lfloor \log_n(n^3 - 1) \rfloor$
for $unit \leftarrow 1$ to $k$ do
    Prepare $n$ empty list $L_0, L_1, \cdots, L_9$
    while $A[]$ isn't empty do
        $temp \leftarrow L.first$
        remove $L.first$ in $L$
        $i \leftarrow$ the $unit^{th}$ number of $temp$
        $L_i.insert(temp)$
    end while
    for $i \leftarrow 0$ to $9$ do
        $L.append(L_i)$
    end for
end for
return $L$
```