1. My Answer:

DFS:

	A	B	C	D	A	B	D
A	1	1	1	1	5	1	1
B	0	2	0	0	0	6	0
C	0	0	3	0	0	0	3
D	0	0	0	4	0	0	7

end state

through KMP ⇒

A	1
B	2
C	3
A	1
B	2
C	3
D	4
A	5
B	6
C	3
D	4
A	5
B	6
D	7

end state

2. My Answer:

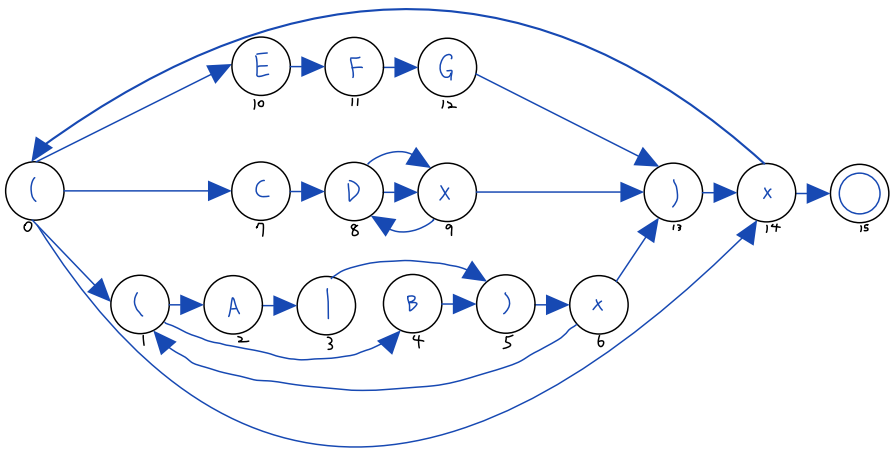
2.1 (a|b|c|d|e)

2.2 (ac)+ equivalent to (ac)(ac)+

2.3 (ab){3,5} equivalent to abab(ab|abababab)

2.4 a[b-d] equivalent to a(b|c|d)

3. My Answer:



	possible states
begin	0, 1, 2, 4, 7, 10, 14, 15
A	0, 1, 2, 3, 4, 7, 10, 14, 15
B	0, 1, 2, 4, 5, 7, 10, 14, 15
B	0, 1, 2, 4, 5, 7, 10, 14, 15
A	0, 1, 2, 3, 4, 7, 10, 14, 15
C	0, 1, 2, 4, 7, 8, 10, 14, 15
E	11
F	12
G	0, 1, 2, 4, 7, 10, 13, 14, 15
E	11
F	12
G	0, 1, 2, 4, 7, 10, 13, 14, 15
C	0, 1, 2, 4, 7, 8, 10, 14, 15
A	0, 1, 2, 3, 4, 7, 10, 14, 15
A	0, 1, 2, 3, 4, 7, 10, 14, 15
B	0, 1, 2, 4, 5, 7, 10, 14, 15
end	15

4. My Python code:

```python
# Hsuan-You Lin Module 11 Problem Set Question 4.
def maxSubArray(nums):
    dp = [0 for i in range(len(nums))]
    dp = nums[0]
    dp1 = 0
    dp2 = dp
    for i in range(0, len(nums)):
        if dp < dp2:
            dp = dp2
            dp_max_id = i
        if dp2 < dp1:
            dp1 = dp2
            dp_min_id = i
        dp2 += nums[i]

    subsequence = nums[dp_min_id : dp_max_id + 1]
    total = sum(nums[dp_min_id : dp_max_id + 1])

    return subsequence, total

if __name__ == "__main__" :
    nums = [5, 15, -30, 10, -5, 40, 10]
    ans = maxSubArray(nums)
    print("Input: ", nums)
    print("Output: ", ans[0])
    print("Sum of contiguous subsequence: ", ans[1])
```

Module11 — -bash — 80×24

```
(base) pisces:Module11 pisces$ python Q4.py
Input:  [5, 15, -30, 10, -5, 40, 10]
Output:  [10, -5, 40, 10]
Sum of contiguous subsequence:  55
(base) pisces:Module11 pisces$
```

5.1. My Python code:

Q5 〉No Selection

```python
# Hsuan-You Lin Module 11 Problem Set Question 5.1.
def Q5_1(wordlist,word):
    dic = {}
    for i in range(len(wordlist)):
        for j in range(i, len(wordlist)):
            dic[wordlist[i: j+1]] = dic.get(wordlist[i: j+1], 0) + 1

    if word in dic:
        return True

    return False

if __name__ == "__main__" :
    wordlist = "itwasthebestoftimesitwastheworst"
    word = "was"
    ans = Q5_1(wordlist, word)
    print("Input: ", wordlist)
    print("Find: ", word)
    print("Output: ", ans)
```

Module11 — -bash — 80×24

```
(base) pisces:Module11 pisces$ python Q5.py
Input:  itwasthebestoftimesitwastheworst
Find:  was
Output:  True
(base) pisces:Module11 pisces$
```

5.2. My Python code:

```python
def dict(word):
    dictionary = [ "it", "was", "the", "best", "of", "times", "it", "it",
        "was", "the", "worst"]
    size = len(dictionary)
    for i in range(size):
        if (dictionary[i] == word):
            return True

    return False

def Q5_2(worldlist):
    dp = [0 for i in range(len(worldlist))]
    dp_index = len(worldlist)-1
    dp1_index = []
    ans = []
    for i in range(len(worldlist)):
        for j in range(i):
            if dict(worldlist[: i+1]):
                dp[i] = i
            if dp[j] != 0 and dict(worldlist[j+1: i+1]):
                dp[i] = j

    while dp_index != dp[dp_index]:
        dp1_index = dp[dp_index]
        ans.append(worldlist[dp1_index+1: dp_index+1])
        dp_index = dp1_index
    ans.append(worldlist[: dp_index+1])
    ans.reverse()

    return ans

if __name__ == "__main__" :
    wordlist = "itwasthebestoftimesitwastheworst"
    ans = Q5_2(wordlist)
    print("Input: ", wordlist)
    print("Output: ", ans)
```

Module11 — -bash — 103×24

```
(base) pisces:Module11 pisces$ python Q5.py
Input:  itwasthebestoftimesitwastheworst
Output:  ['it', 'was', 'the', 'best', 'of', 'times', 'it', 'was', 'the', 'worst']
(base) pisces:Module11 pisces$
```

6. My Python code:

```python
# Hsuan-You Lin Module 11 Problem Set Question 6.
import sys

def Q6(arr, start, length):

    count = 0

    if start == length:
        return 0

    result = sys.maxsize

    for i in range(start, length):
        count = (Q6(arr, start, i)
                 + Q6(arr, i + 1, length)
                 + arr[start-1] * arr[i] * arr[length])

        if count < result:
            result = count

    return result

if __name__ == '__main__':
    arr = [10, 10, 10, 1]
    # 3 matrices of dimensions 10x10, 10x10, 10x1
    ans = Q6(arr, 1, len(arr)-1)
    print("Minimum number of multiplications is ", ans)
```

Module11 — -bash — 80×24

```
(base) pisces:Module11 pisces$ python Q6.py
Minimum number of multiplications is  200
(base) pisces:Module11 pisces$
```

7. My Answer:

```python
# Hsuan-You Lin Module 11 Problem Set Question 7.
def Q7(str1, str2, i, j):
    if i == 0:
        return j

    if j == 0:
        return i

    if str1[i-1] == str2[j-1]:
        return Q7(str1, str2, i-1, j-1)

    return 1 + min(Q7(str1, str2, i, j-1),    # Insert a char
                   Q7(str1, str2, i-1, j),    # Delete a char
                   Q7(str1, str2, i-1, j-1)    # Change 1 char into another char
                   )

if __name__ == "__main__" :
    str1 = "cast"
    str2 = "cats"
    ans = Q7(str1, str2, len(str1), len(str2))
    print("Answer is: ", ans)
```

Module11 — -bash — 80×24

```
(base) pisces:Module11 pisces$ python Q7.py
Answer is:  2
(base) pisces:Module11 pisces$
```