COS 226 Data Structures and Algorithms
Computer Science Department
Princeton University
Fall 2015

Week 11 handout

1. **Data Compression**

   (a) **Burrows-Wheeler transform**

   i. What is the Burrows-Wheeler transform of b a n a n a?

```
suffix[0] = b a n a n a
suffix[1] =
suffix[2] =
suffix[3] =
suffix[4] =
suffix[5] =

Sorted Suffixes
suffix[0] =
suffix[1] =
suffix[2] =
suffix[3] =
suffix[4] =
suffix[5] =
```

   Write your answer in the box.

   ii. Apply the Burrows-Wheeler inverse transform to find the original string

```
6
t[] = helweer
```

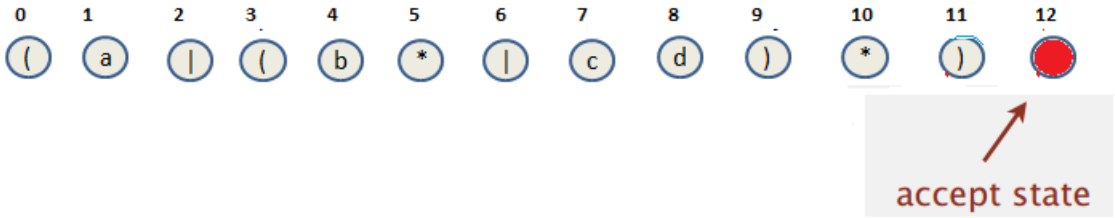
   Construct the next array as shown in the Burrows-Wheeler assignment and find the original string.

i	sorted suffixes	t	next
0			
1			
2			
3			
4			
5			
6			

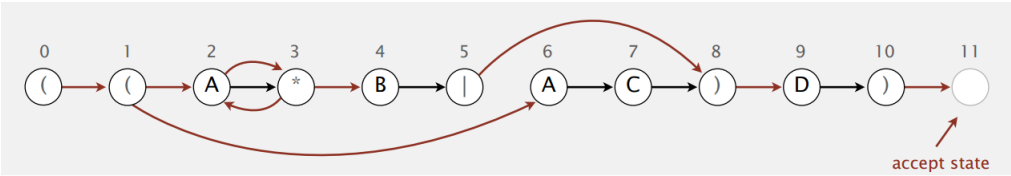
Write your answer in the box.

2. **Regular expressions and Non-Deterministic Finite Automata(NFA)**

(a) Convert the regular expression $(a \mid (b* \mid cd)*)$ into an equivalent NFA using the algorithm described in lecture by adding black edges and ϵ transition edges to the diagram below.



(b) Which of the following strings are accepted by the NFA given below? For a string that is accepted show how the machine transitions get to the accept state and when a string is not accepted, show that machine transitions never get to the accept state. You can take epsilon transitions and can be in multiple states before character A is scanned.



i. AB
ii. BD
iii. $AAAAAB$
iv. ACD

3. **LZW Compression (Bonus Problem)**
Assume that we are working with the ASCII alphabet where a=61, b=62 in hexadecimal. The end of file character is 80. The next available code is 81.

(a) Encode the message **abbbabba** using LZW compression. Fill in the following table using new codes discovered

symbol	code

Write the encoded message in the box.

(b) You receive the following LZW encoded message. Decode the message. Note that decoding this message involves the tricky case, where you see a code, before it is in the table. Hence you need to construct the missing code from prior knowledge.

```
61 62 81 83 62 80
```