

Quiz 6

Write your name on the exam. Write something for every question. Students who do not write something for everything lose out over students who write down wild guesses. You will get some points if you attempt a solution but nothing for a blank sheet of paper. Write something down, even wild guesses. Problems take long to read but can be answered concisely.

Question	Maximum	Score
1	11	
2	9	
Total	20	

Question 1 – Compression

Give an algorithm for run length encoding that requires only a single byte to encode (most) non-repeated symbols. Keep in mind that decoding must be unambiguous for any possible input to the compression algorithm. Within this constraint the more compact the compressed string, the better. There are multiple correct solutions, and no solution gives better compression than all the others on all possible inputs.

This algorithm is related to the escape code algorithm we discussed in class. It produces slightly smaller compressed data for some inputs. The encoding rule is as follows: non-repeating characters are encoded as themselves, repeating characters are encoded as two repetitions of the character followed by a count. Since single characters do not repeat (by definition) this encoding is unambiguous. Example:

AAAAAABCDDDEEF1A6HH8 => AA7BCDD3EE2F1A6HH28

This encoding is unambiguous. Note that the repeat counts are always at least 2, so we can encode repetition counts from 2 to 257 in a single byte. If the number of repetitions is larger than 257, we encode the first 257 and treat the rest separately.

```
Encode()
while(c=getNextCharacter()){
    counter=1
    while(nextAfterCurrent()==c && counter<257){
        getNextCharacter();
        counter=counter+1;
    }
    if(counter==1){
        Output(c);
    }else{
        Output(c,c,char(counter-2));
    }
}
```

```
Decode()
while(c=getNextCharacter()){
    if(nextAfterCurrent()!=c){
        Output(c);
    }else{
        getNextCharacter();
        counter=int(getNextCharacter()+2);
        for(int i=0;i<counter;i++){
            Output(c);
        }
    }
}
```

Question 2 – email

a) DNS and the web rely heavily on caching to improve performance. Why doesn't email also use caching to improve performance?

Caching works for web documents and DNS records because the server hosting them is far and they are read repeatedly. Since the final mail server is typically close to the mail agent of the receiver, caching is not very useful for email.

b) Why do SMTP servers support authenticating clients that connect to them to send email to remote users?

By authenticating clients, SMTP servers can make sure that they are not relaying spam (unless the computer of a legitimate client got hijacked by spammers).

c) Assuming a web based email service is implemented by extending the functionality of a web server. Would the software implementing this service have to implement the same functionality as a mail server or as a mail agent?

Mail agent.