CSIT 113 (Problem Solving)

Assignment 1

Name: Mohamed Haneefa Jiyavudeen

Tutorial group: T03F

UOW number: 8496407

Question 1)

1. 4! = 4 x 3 x 2 x 1 = 24
2. 5! = 5 x 4 x 3 x 2 x 1 = 120
3. 6! = 6 x 5 x 4 x 3 x 2 x 1 = 720
4. n! = n x (n-1) x (n-2) x … x 3 x 2 x 1

Question 2)

1. Number of total possible outcomes (without constraint)

= 4^4

= 256

1. Number of total possible outcomes (with constraint)

= 4!

= 4 x 3 x 2 x 1

24

1. Legend:

Freestyle 🡪1

Breaststroke 🡪2

Butterfly 🡪3

Backstroke 🡪4

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Andrew** | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 3 | 3 | 3 | 3 | 3 | 3 | 4 | 4 | 4 | 4 | 4 | 4 |
| **Bernard** | 2 | 3 | 2 | 4 | 4 | 3 | 1 | 3 | 1 | 4 | 4 | 3 | 1 | 2 | 1 | 4 | 4 | 2 | 1 | 2 | 1 | 3 | 3 | 2 |
| **Chester** | 3 | 2 | 4 | 2 | 3 | 4 | 3 | 1 | 4 | 1 | 3 | 4 | 2 | 1 | 4 | 1 | 2 | 4 | 2 | 1 | 3 | 1 | 2 | 3 |
| **Daniel** | 4 | 4 | 3 | 3 | 2 | 2 | 4 | 4 | 3 | 3 | 1 | 1 | 4 | 4 | 2 | 2 | 1 | 1 | 3 | 3 | 2 | 2 | 1 | 1 |
| Outcomes | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 |

1. All combinations

19 – 172

20 – 175

21 – 172

22 – 169

23 – 174

24 - 180

13 – 173

14 – 176

15 – 172

16 – 167

17 – 172

18 - 180

7 – 177

8 – 174

9 – 176

10 – 171

11 – 176

12 - 178

1 – 183

2 – 177

3 – 182

4 – 174

5 – 174

6 - 176

Best combination for optimal time is:

Andrew – Butterfly ( 3 )

Bernard – Backstroke ( 4 )

Chester – Freestyle ( 1 )

Daniel – Breaststroke ( 2 )

Question 3)

1. { A B C D } { C D | A B , 🡪 | } { CD | | AB } : Initially all at left bank, then, **Alice** and **Bob** row over to right bank, now, **Carol** and **Dave** at left bank and **Alice** and **Bob** at right bank
2. { CD | | AB } { C D | A , 🡨 | B } { ACD | | B } : **Carol** and **Dave** at left bank, **Alice** row from right bank to left bank, and now **Alice**, **Carol** and **Dave** at left bank and **Bob** at right bank
3. { ACD | | B } { D | AC , 🡪 | B } { D | | ABC } : **Alice**, **Carol** and **Dave** at left bank, **Alice** and **Carol** row to right bank, and now **Alice**, **Bob** and **Carol** at right bank while **Dave** is at left bank.
4. { D | | ABC } { D | A , 🡨 | BC } { AD | | BC } : **Dave** is at left bank, **Alice** row back to left bank, and now, **Alice** and **Dave** at left bank, while **Bob** and **Carol** at right bank.
5. { AD | | BC } { A | D , 🡪 | BC } { A | | BCD } : **Alice** at **Dave** at left bank, then **Dave** row to right bank leaving **Alice** at left bank and **Bob**, **Carol** and **Dave** at right bank.
6. { A | | BCD } { A | C , 🡨 | BD } { A C | | BD } : Since **Alice** is at left bank, **Carol** has to row back from right bank to left bank. Now **Alice** and **Carol** at left bank, **Bob** and **Dave** at right bank.
7. { AC | | BD } { | A C , 🡪 | BD } { | | ABCD } : **Alice** and **Carol** row from left bank to right bank and now everyone has crossed over to right bank safely. **Alice**, **Bob**, **Carol** and **Dave** are at right bank and no one is at left bank.

Minimum number of **rows is 7**.

Question 4)

p(q  r)  ((pq)  (pr))

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| p | q | r | (q  r) | p(q  r) | pq | pr | ((pq)  (pr)) | p(q  r)  ((pq)  (pr)) |
| T | T | T | T | T | T | T | T | T |
| T | T | F | F | F | T | F | F | T |
| T | F | T | F | F | F | T | F | T |
| F | T | T | T | F | F | F | T | F |
| F | F | F | T | F | F | F | T | F |
| F | F | T | F | F | F | F | T | F |
| F | T | F | F | F | F | F | T | F |
| T | F | F | T | T | F | F | T | T |

In this case, p(q  r)  ((pq)  (pr)) is only true under 4 scenarios and is false under 4 scenarios. Hence p(q  r)  ((pq)  (pr)) is not always true.

Question 5)

(p  q)(q  r)  (p  r)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| p | q | r | p  q | q  r | p  r | (q  r)  (p  r) | (p  q)(q  r)  (p  r) |
| T | T | T | T | T | T | T | T |
| T | T | F | T | F | F | T | T |
| T | F | T | F | T | T | T | T |
| F | T | T | T | T | T | T | T |
| F | F | F | T | T | T | T | T |
| F | F | T | T | T | T | T | T |
| F | T | F | T | F | T | T | T |
| T | F | F | F | T | F | F | T |

Hence it is proven that the formula is always true for (p  q)(q  r)  (p  r)

Question 6)

1. Given, A  (B  C) and B  ~ C

Substitute B  ~ C into A  (B  C)

A  (~ C  C)

Since (~ C  C), C is not true.

If C is not true, then C is a knave, hence proves that Bob is a knight.

And Alice says that Bob and Carol are the same type.

Since we know Bob is a knight and Carol is a knave, we can conclude that **Alice is a knave.**

1. Let J represent the proposition “John is a knight”.
2. Let S represent the proposition “Steven is a knight”.
3. Let Q represent the proposition “The question to ask steven to identify exactly what John is”

(S  Q)  J

Q  (S  J)

Therefore, the question to ask Steven is “Is John a knight?”

If Steven say yes, it proves that John is a knight as we have assumed Steven is saying the truth. If Steven say no, then we know that John is a knave assuming that Steven is saying the truth.

Question 7)

Recursive Algorithm for

SumOfR(A,n) {

for n ≥ 1

if n = 1

sum = 1/A[0]

else

sum = SumOfR(A,n-1) + 1/A[n-1]

Question 8)

A paper with writing on it

Description automatically generated