Question1

T

a. The snowflake schema model saves storage space compared to the star schema model

T

b. OLAP systems usually adopts an entity-relationship data model.

T

c. Standard deviation can be derived by applying the function to n aggregate values in a data cube.

F

d. One benefit of the Multiway Array Aggregation cube is the equal efficiency regardless of scan order.

F

e. Unlike the Multiway Array Aggregation method, BUC cannot take advantage of pruning.

Question2

```
(b1; a2; a3; a4; a5); (a1; b2; a3; a4; a5); (a1; a2; b3; a4; a5)
```

a. (5, L2) How many cuboids are there in the full data cube?

There were 5 dimensions, and First 3 each have 2 cells.

It would be ***(2+1)*****(2+1)*****(2+1)*****(1+1)*****(1+1) = 108*** Cube

b.How many nonempty aggregate cells are there in the full cube?

```
Since all 3 base cell have a4, a5, we can enumulate first 3:
b1,*; a2,*; a3,*;
a1,*; b2,*; a3,*;
a1,*; a2,*; b3,*;

it would be 2*2*2*3 = 24

For all 3, they have (*;*;*;), so 24 - 2 = 22

For each 2 row (like 1st,2nd) have one same cube (like (*;*;a3;))
22 - 1*3 = 19
```

Then we combine the a4, a5

19*(1+1)*(1+1) = 76

If we did not calculate base cell:

76 - 3 = ***73***

c.How many nonempty aggregate closed cells in the full cube

Since we do not need to consider the aggregate cell,it will be ***4*** as follow:

```
(a1;  *;  *; a4; a5);
(*;  a2;  *; a4; a5);
( *;  *; a3; a4; a5);
( *;  *;  *; a4; a5);
```

d. How many nonempty aggregate cells an iceberg cube contain

if the condition of the iceberg cube is count >= 3?

it will be ***4*** as follow

```
( *;  *;  *; a4; a5);
( *;  *;  *; *; a5);
( *;  *;  *; a4; *);
( *;  *;  *;  *; *);
```

Question3

3.a

From 1 to 4, B0C0 will be filled, BC need 1 memory chunk, AB,AC need 4 memorie chunks;

From 1 to 16, A0B0,A0B1,A0B2,A0B3 will be filled, AC need 16 memorie chunks.

Since the sizes of the chunks on dimensions A, B, and C are 25, 50, and 50, chunk in AB need 1250, AC need 1250, BC need 2500.

The total memory need for holding 2D plane is:

In [1]:
```
50*50*1 + 25*50*4 + 25*50*16
```

Out[1]:
```
27500
```

3.b

```
Yes, an alternative way to get the eqaul minimum demand memory is to scan in
the order:

1,2,3,4;17,18,19,20;...
5,6,7,8;21,22,23,24;...
...
...;61,62,63,64;
```

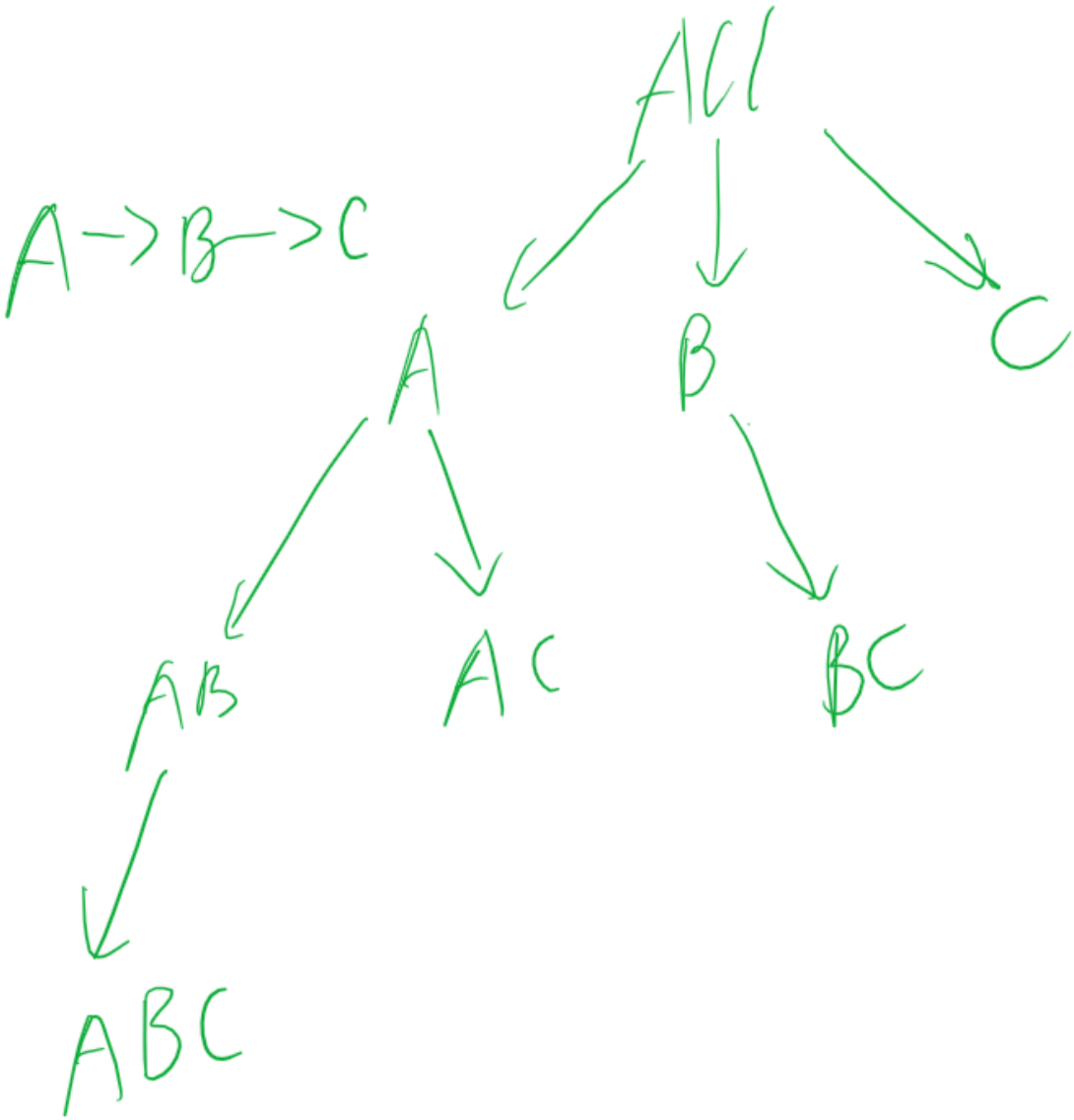
From 1 to 4, B0C0 will be filled, BC need 1 memory chunk, AB,AC need 4 memorie chunks;

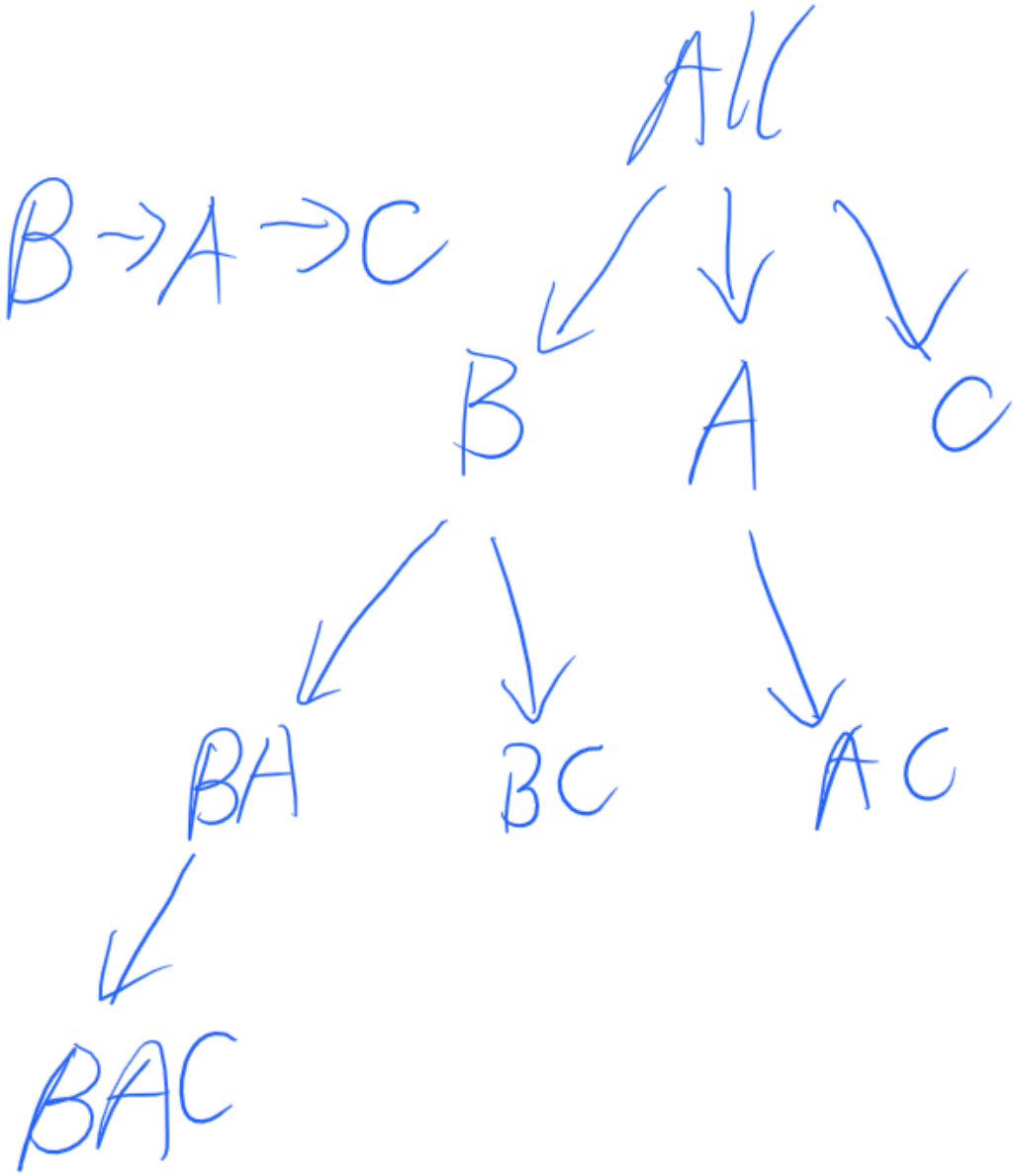
From 1 to 52, A0C0,A0C1,A0C2,A0C3 will be filled, AB need 16 memorie chunks.

In this way, the minmum total memory is the same

Question4

4.a


A->B->C
ACI
A
B
C
AB
AC
BC
ABC



4.b

```
If we follow the A -> B -> C:
All (*,*,*) : 18 - expansion
 A (a0,*,*) : 18 - expansion
   AB(a0,b0,*) : 3
   AB(a0,b1,*) : 6
   AB(a0,b2,*) : 9 - expansion
        ABC(a0,b2,c0) : 3
        ABC(a0,b2,c1) : 3
        ABC(a0,b2,c2) : 3
   AC(a0,*,c0) : 6
   AC(a0,*,c0) : 6
   AC(a0,*,c0) : 6
 B (*,b0,*) : 3
 B (*,b1,*) : 6
 B (*,b2,*) : 9 - expansion
   BC(*,b2,c0) : 3
   BC(*,b2,c1) : 3
   BC(*,b2,c2) : 3
 C (*,*,c0) : 6
 C (*,*,c1) : 6
 C (*,*,c2) : 6
```

Thus, there are totally ***20*** cells which would have to be computed

4.c

```
If we follow the C -> B -> A:
All (*,*,*) : 18 - expansion
 C (*,*,c0) : 6
 C (*,*,c1) : 6
 C (*,*,c2) : 6
 B (*,b0,*) : 3
 B (*,b1,*) : 6
 B (*,b2,*) : 9 - expansion
   AB(a0,b0,*) : 3
   AB(a0,b1,*) : 6
   AB(a0,b2,*) : 9
 A (a0,*,*) : 18
```

Thus, there are totally ***11*** cells which would have to be computed