1)

c)

When we combine the summaries of two sublists, we need these information.

- The right end of the first sublist
- The left end of the second sublist
- Biggest distance tuple {distance, index}.
- the number of vectors in a sublist

2)

b) The node need these information.

- the key
- the count of each keys

The information is useful for both the upward and downward pass.

i)

Time complexity:

running_count_ser() : O(n)
rcp_leaf1(): O(N/P)
rcp_combine(): O(K)
rcp_root(): O(1)
rcp_leaf2(): O(N/P)
running_count_par() : O(N/P+KlogP)

Space complexity:
running_count_ser() : O(nK)
ProcStat : O(NK/p)
Message : O(K)

j)

P = 32 , N= 100 ,M = 100
SeqTime = 0.000017
ParTime = 0.000223
SpeedUp = 0.07718593632302209
P = 32 , N= 1000 ,M = 100
SeqTime = 0.000197
ParTime = 0.000600
SpeedUp = 0.32796596668490197
P = 32 , N= 10000 ,M = 100
SeqTime = 0.001969

ParTime = 0.001513
SpeedUp = 1.3013933174274206
P = 32 , N= 100000 ,M = 100
SeqTime = 0.026656
ParTime = 0.009354
SpeedUp = 2.8495236689843884
P = 32 , N= 1000000 ,M = 100
SeqTime = 0.594591
ParTime = 0.092806
SpeedUp = 6.406842831542136

P = 1 , N= 1000000 ,M = 100
SeqTime = 0.496833
ParTime = 1.125992
SpeedUp = 0.4412403404826895
P = 2 , N= 1000000 ,M = 100
SeqTime = 0.546545
ParTime = 0.832784
SpeedUp = 0.6562874788646607
P = 4 , N= 1000000 ,M = 100
SeqTime = 0.448665
ParTime = 0.445064
SpeedUp = 1.0080912346481727
P = 8 , N= 1000000 ,M = 100
SeqTime = 1.331783
ParTime = 0.219897
SpeedUp = 6.056384677086701
P = 16 , N= 1000000 ,M = 100
SeqTime = 0.504913
ParTime = 0.129351
SpeedUp = 3.9034462433616524
P = 32 , N= 1000000 ,M = 100
SeqTime = 0.502153
ParTime = 0.084394
SpeedUp = 5.950099623735606
P = 64 , N= 1000000 ,M = 100
SeqTime = 1.440926
ParTime = 0.086400
SpeedUp = 16.677370836071482
P = 128 , N= 1000000 ,M = 100
SeqTime = 0.686327
ParTime = 0.093082
SpeedUp = 7.37339492257531

P = 256 , N= 1000000 ,M = 100
SeqTime = 0.532602
ParTime = 0.177314
SpeedUp = 3.0037261000843785

P = 32 , N= 100000 ,M = 10
SeqTime = 0.008829
ParTime = 0.004439
SpeedUp = 1.9888624406510655
P = 32 , N= 100000 ,M = 100
SeqTime = 0.021231
ParTime = 0.009448
SpeedUp = 2.2471259083661184
P = 32 , N= 100000 ,M = 1000
SeqTime = 0.042099
ParTime = 0.023542
SpeedUp = 1.7882322385456852
P = 32 , N= 100000 ,M = 10000
SeqTime = 0.044251
ParTime = 0.074270
SpeedUp = 0.5958033453131196

My sequential version runtime is running in O(n). When the data size increase and its runtime also increase. My parallel version runtime is running in O(N/P+KlogP).It can be observed that the speed up is increasing when we fix the number of processes and increases the number of data. This corresponds to the models in class that when the number of data increases, the computation cost dominates the communication cost.

In addition, it can be observed that the speed up is increasing when we fix the number of data and increases the number of processes, but the speed up decreases slightly when the processes reaches 256. This also corresponds to the models in class that when the number of processes increases, the computation cost dominates the communication cost, but when there are too many processes, the communication cost increases again.

For sequential version, the runtime increase as the M increase, which does not agree with the theoretical analysis. This should relate to space complexity. For parallel version, M is equal to K, when K is increasing , the Message size will also increase. And the communication part will dominate the function. And the speedup will decrease with M size increase.

3)

a)

P is perfect square and it is a torus cut it into half. Therefore, the cross-section bandwidth is $2\sqrt{P}$ .

b)

We have N/2 data need to pass through cross-section. And the cross-section bandwidth is $\frac{N}{2(2\sqrt{P})}$ .

Therefore , the lower bound is $\Omega(\frac{N}{\sqrt{P}})$.

c)

$P \leq (\text{LogN})^2$

d)

P is perfect cube and it is a torus cut it into half. Therefore, the cross-section bandwidth is 2P^(⅔).

e)

The lower bound is $\Omega(\frac{N}{P\hat{}(2/3)})$.

f)

$P \leq (\text{LogN})^3$

g)

3c: $(\log 2^{30})^2 =$ 900

3f: $(\log 2^{30})^3$ 27000