Question 1

a.
In map phase, each record (A, B) will be checked against the predicate “B >= 1000 and B < 2000”. Those records that do not satisfy the predicate will get discarded. Other records will be sent to reducers with A as the key and B as the value.

In reduce phase, do one scan of all the values associated with one key and pick the maximum one, output record with A as the key and the maximum value as the value.

Combiner, which is the same as reducer, could be used at map side to reduce the amount of data being shuffled.

b.
In this scenario, one mapper will only sent its intermediate result to one reducer and mapper 1, mapper 5-10 has no record satisfying the predicate.

If combiner is not used, mapper 2 outputs 2 records to reducer 1, mapper 3 outputs 1000 records to reducer 1 and mapper 4 outputs 998 records to reducer 1. Others have no output to reducers.

If combiner is used, mapper 2 output 1 record to reducer 1, mapper 3 outputs 250 records to reducer 1 and mapper 4 outputs 250 records to reducer 1. Others have no output to reducers.

Question 2.

1. equivalent
2. inequivalent. If S or T has a duplicate value of A that will match a record in R, then the final result will also have duplicate values of A, which will never happen in plan (a). If all the matching records in S and T have no duplicate A, the result is equivalent to plan (a)
3. inequivalent. Similar to question 2, if all the matching records in T has no duplicate A, the result is equivalent to plan (a).
4. equivalent

Question 3.

1. to do the first TNLJ, the number of getNext() on scan of R and S is 6+6*(2+1)+1=25, which generate 4 records as intermediate result. To do the second TNLJ, the number of getNext() on the scan of intermediate result and T is 4+4*(12+1)+1=57, which generates 12 records. To do the final projection, 12+1=13 getNext() is needed. There are 2+1 calls for the final output of the query. The total number of getNext() is 25+57+13+3=98.
2. The first TNLJ requires 12+12*(2+1)+1=49 getNext() on scan of T and S and generates 6 records. The second TNLJ requires 6+6*(6+1)+1=49 getNext() on the scan of intermediate result and R and generates 12 records. The final projection requires 12+1=13 getNext(). There are 2+1 calls for the final output of the query. The total number of getNext() is 49+49+13+3=114.
3. the projection on R.A requires 6+1=7 getNext() on scan of R and generates 3 records. The first
TNLJ requires $3+3*(2+1)+1=13$ getNext() and generates 2 records. The second TNLJ requires $2+2*(12+1)+1=29$ getNext() and generates 6 records. The final projection requires $6+1=7$ getNext(). There are $6+1$ calls for the final output of the query. The total number of getNext() is $7+13+29+7+7=63$.

4. the first TNLJ requires $6+6*(2+1)+1=25$ getNext() and generates 4 records. The projection on the intermediate result requires $4+1=5$ getNext(). The second TNLJ requires $2+2*(12+1)+1=29$ getNext() and generates 6 records. The final projection on this requires $6+1=7$ getNext(). There are $6+1$ calls for the final output of the query. The total number of getNext() is $25+5+29+7+7=73$.

5. The first TNLJ on R and S requires $6+6*(2+1)+1=25$ getNext() and generates 4 records. The projection on this requires $4+1=5$ getNext() and generate 2 records. The entire inner part of the outermost join will get called twice. Hence, there will be $2*(12+1)$ calls to Scan(T) and generates 4 records. The second TNLJ on these two parts requires $2+2*(4+1)+1=13$ getNext() and generates 2 records. The final projection needs $2+1=3$ getNext(). There are $2+1$ calls for the final output of the query. The total number of getNext() is $25+5+26+13+2+3=75$.

**Question 4.**
the basic structure of the left-most-deep tree likes like this

Consider the following 3 situations:

1. When A is duplicate-eliminate projection. B, C, D and E could be duplicate-eliminating projection or duplicate-preserving projection or nothing. There are $3*3*3*3=81$ combinations.
2. When A is duplicate-preserving projection while B and C are duplicate-eliminating projections. D and E could be duplicate-eliminating projection or duplicate-preserving projection or nothing. There are $3*3=9$ combinations.
3. When A is duplicate-preserving projection while C, D and E are duplicate-eliminating projections. B could be duplicate-preserving projection or nothing. There are 2 combinations.

The number of different order of R, S and T is $3!=6$. There are two type of tree (left-deep-most and right-deep-most). So the total number of physical plans is $2*6*(81+9+2)=1104$

**Question 5.** (solution from Yi Zhang)
To get the minimum number of getNext() calls, we want to put T in the left most position so that we won't produce $n(12+1)$ getNext() calls where $n>1$. 
For TNLJ, we always want to relation with smaller cardinality on the left. So we come up with the following plan:

\[
\begin{align*}
\text{Scan}(S) \\
\text{Scan}(R) \\
\Pi_{R.A} \\
\Pi_{T.A} \\
\text{Scan}(S) \\
\text{Scan}(T)
\end{align*}
\]

the number of getNext() calls is:
\[
(12+1)+(4+1)+4*(2+1)+(2+1)+2*(6+1)+(4+1)+(2+1)=55
\]