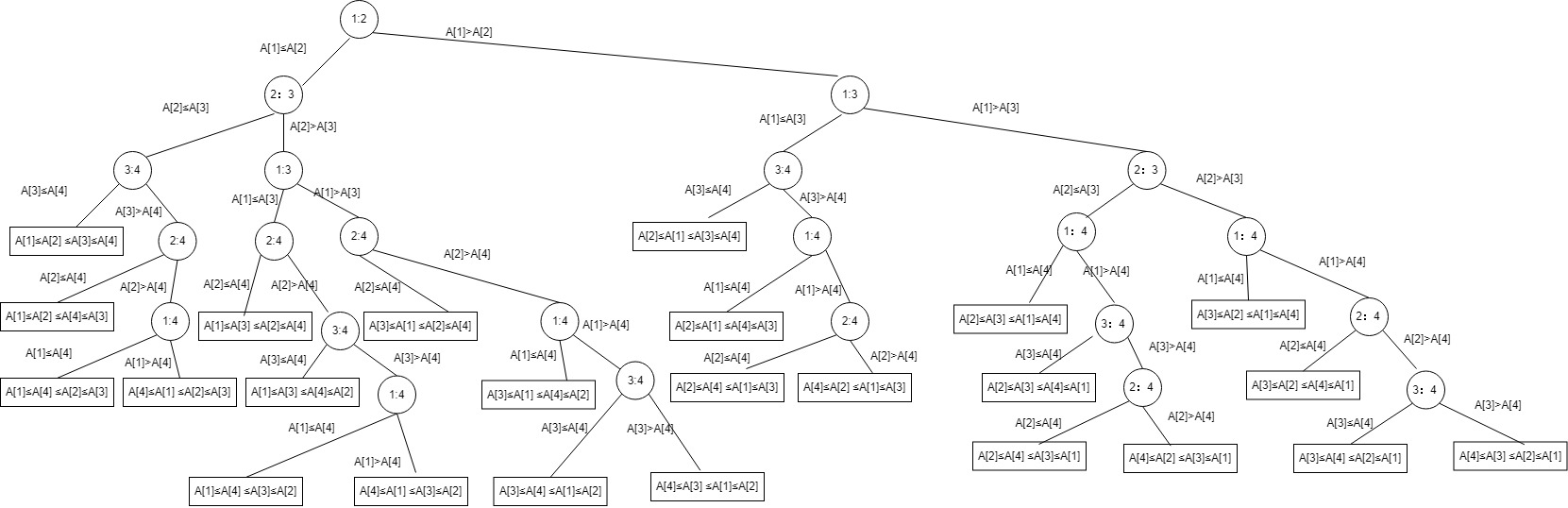
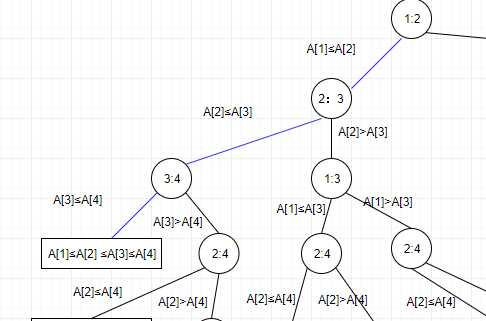
1A. The decision tree for selection sort with 4 elements (may need a little bit zoom in)

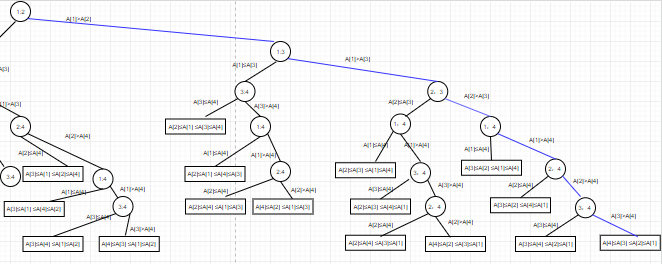


2A. Best path A[1]≤A[2]≤A[3]≤A[4]



In selection sort, this means if the given array is already in ascending order then the decision tree will have shortest path.

3A. Worst path A[4]≤A[3]≤A[2]≤A[1]



In selection sort, this means if the given array is already in descending order then the decision tree will have worst path.

4A. I have 24 leaves from the decision tree I draw, and this means with 4 elements in selection sort, it has 24 comparison combinations. In the definition we have each possible sorted order corresponds to a permutation, so there are at least n! leaves.

2. By the definition of the time complexity of Radix sort, Given n b-bit numbers and any positive integer r≤b, Radix-sort will sorts these numbers in Θ((b/r)(n+2r))time if the stable sort is uses takes Θ(n+k) time for inputs in the range 0 to k.

So for a value r≤b, we know each key having d=[b/r] digits with r bits each. And each digit is an n integer in the range of 0 to 25-1, so that means r=5. In this example, n is integer which has 16 bits, so the base of the integer is 16. B=16. but we have to calculate value of d, which is the number of digits for this integer. So r=lg(25-1)=2 bits. Then the total time complexity of Radix sort is Θ((16/2)(n+4)) = **Θ(8(n+4))**