# Take home exam 1

Kailiang Ying

## 1 Description of algorithm:

#### First step: Initial some parameter :

1. Randomly choose one value in DVD and give value to **pivot**. Store **pivot** in memory

2. Store three parameters in memory as counter ,CL ,CR ,CE ,initial all zero

**CL** : count the number of value less than pivot in DVD

**CR** : count the number of value greater than pivot in DVD

**CE** : count the number of value equal to pivot in DVD

3. **prep** store the previous chosen pivot

4. **temp** store one value from DVD

so there will be six parameter in memory

#### Second step: start compare:

1. While DVD has value not been read ,keep read one value and store in temp.
2. compare temp with pivot
3. if temp is greater than pivot ,CR increase one
4. if temp is smaller than pivot ,CL increase one
5. if temp is equal to pivot ,CE increase one

#### Third step: choose pivot :

1. if CR>CL+CE

then if pivot <prep then prep = pivot ,pivot randomly choose one value from DVD greater than pivot and less than prep

else if pivot >prep then prep =pivot , pivot randomly choose one value from DVD greater than pivot

1. if CL>CR+CE

then if pivot>prep then prep =pivot , pivot randomly choose one value from DVD greater than prep and less than pivot

else if pivot <prep ,then prep =pivot ,pivot randomly choose one value from DVD less than prep

1. if number of value in DVD is odd

if CR<CL+CE and CL<CR+CE

return pivot as median

4. if number of value in DVD is even

if C if CR<CL+CE and CL<=CR+CE

return pivot as median

5. If have not find the median , back to first step

#### Pseudo –code :

function Random

input : list of value ,CL,CR,CE

output: randomly choose pivot

function Select

input: list of value ,pivot

output: median

code:

pivot = DVD[0] //initial pivot be the first value of DVD

prep =pivot //initial prep equal to pivot

CR=0; CL=0; CE=0; //initial three counter zero

temp=0;

//function random produce random number to pivot

random(DVD,CL,CR,CE)

{

//choose from the greater number

if (CR>CL+CE)

if(pivot > =prep)

pivot randomly choose from value in DVD

if(pivot <prep)

pivot randomly choose from value in DVD

//choose from the smaller number

if(CL>CR +CE)

if(pivot >prep)

pivot randomly choose from value in DVD

if(prep<=pivot)

pivot randomly choose from value in DVD

}//end function Random

//function to find the median

Select(DVD, pivot)

{

//loop counter

J=0;

//three counter

CL=0;

CR=0;

CE=0;

while( DVD[J] is not out of bound )

{

temp =DVD[J]

J=J+1

if(temp>pivot)

CR=CR+1;

If (temp<pivot)

CL=CL+1;

If(temp==pivot)

CE=CE+1;

}

if((CL+CR+CE)/2 ==0 ) //the number of value in DVD is even

{

if (CE>1)

If(CR<=CL+CE and CL<=CR+CE) //if the median equal to pivot

Return pivot

If(CE==1)

If(CL+CE=CR)

Return pivot

}

if((CL+CR+CE)/2 !=0) //the number of value in DVD is odd

{

If(CR<CL+CE and CL<CR+CE) //if the median equal to pivot

Return pivot

}

//if not return pivot ,recursive do the choose another pivot

temp =pivot;

pivot =random(DVD,CL,CR,CE);

prep =temp;

temp =0;

Select(DVD, pivot) //recursive do select until return the pivot

}

## 2 proof correctness :

The median is the 50th percentile of the list of value in DVD. If even , choose the smaller of two .

In the algorithm , I use three counters to record the distribute of the list of value. No matter what number in the list , it will correspond to three situation: 1.equal to the pivot ,2 greater than the pivot ,3 less than the pivot.

So every time run Select function, compare all list of value in DVD with pivot (include pivot) , CL+CR+CE = number of value in DVD

After each selection, there exits three situations :

1. CR > CL+CE: means pivot is smaller than median
2. CL > CR+CE: means pivot is larger than median
3. CR <= CL+CE and CL <=CR+CE :means pivot is equal to median when CE>1

Other wise when CE =1 ,

then if number of value is even then CL+CE =CR then pivot is median

if number of value is odd ,then CR < CL+CE and CL < CR+CE then pivot is median

Then the question become how to choose pivot in order to close to median little by little . I use Random function to produce pivot . Random function input CR, CL, CE to create a choosing scope for pivot to choose from the list of value in DVD .

If CR>CL+CE ,means median is greater than pivot ,then see the prep, which store the previous pivot . If prep> pivot , then the choosing scope is (pivot ,prep) .If prep<=pivot ,then the choosing scope is (pivot,)

If CL>CE+CR, means the median is smaller than pivot, then if prep>=pivot chosen scope is (,pivot) . If prep<pivot ,choosing scope is (prep, pivot)

I use this strategy to randomly choose pivot in order to close to median by degree until finally find the median.

For the memory is O(logn) and this algorithm must store 6 parameter in memory in order to work correctly . So if the number of value in DVD is less than 64 (log64 =6), this algorithm does not work.

## 3 Analysis the algorithm :

The worst case is every time pick the largest or the smallest number .So under this situation ,it needs n times to find the median and every time choose pivot needs n times compare .

Choose pivot need O(n) to find randomly pivot

T(n) =n\*n +O(n)=

The best case is choose only one time and just find the median. It also needs n times compare to ensure it is the median.

T(n) =n+ O(n) =O(n)

However, this two case are extremely unlikely to occur every time

So in order to distinguish between lucky choice and unlucky choice , I call 25th to 75th percentile of list value is better choice . I like these choice because this ensure the CR,CL be at most three-fourth of total size of value in DVD.

Fortunately, there are half the elements must fall between 25th to 75th percentile .So that a random chosen pivot has a 50% chance to be good choice and 50% chance to be bad choice.

Proof:

if Probability =0.5 ,then

E =1+0.5E 🡪 E =2

So ,randomly ,I pick up one time good choice before one time bad choice ,and every expect after I time good choice and one time bad choice ,the choosing scope can reduce one-fourth in the worst case.

So the average do times <= 2\* =O(logn)

For this expect , I can half of the good choices and half of the bad choices

T(n)=n\*2\* + O(n)=O(nlogn)