Linear Time Selection

计算机1202 张艺瀚

July 12, 2014

```cpp
#include <algorithm>
#include <cassert>
#include <cstdio>
#include <cstdlib>
#include <ctime>
#include <fstream>
#include <functional>
#include <iostream>
#include <iterator>
#include <limits>
#include <map>
#include <numeric>
#include <regex>
#include <set>
#include <string>
#include <vector>

using namespace std;

int cnt = 0;

void disp(const vector<int>& a)
{
  for(auto& p: a)
    cout << p << " ";
  cout << endl;
}

int my_partition(vector<int>& a, int p, int r, int x)
{
  swap(*find(a.begin(), a.end(), x), a[p]);

  int i = p;
  int j = r;
  while(i < j)
  {
    while(i < j && a[j] >= x) --j;
    a[i] = a[j];
    while(i < j && a[i] <= x) ++i;
    a[j] = a[i];
  }
  a[i] = x;
  return i;
```

```
}

int select(vector<int> a, int p, int r, int k)
{
  ++cnt;
  cout << "call select " << cnt << ":" << endl;

  if(r - p + 1 <= 5)
  {
    cout << "recursion exit:" << endl;
    vector<int> my_a(r - p + 1);
    copy(a.begin() + p, a.begin() + r + 1, my_a.begin());
    cout << "the sub-vector whose length is no greater than 5 is:"
    << endl;
    disp(my_a);
    sort(my_a.begin(), my_a.end());
    cout << "return: " << my_a[p + k - 1] << endl << endl;

    cout << endl << "left select from recursion exit" << cnt <<
    endl;
    return my_a[p + k - 1];
  }

  cout << "find median of medians:" << endl;
  vector<int> medians;
  for(int i = 0; i < (r - p + 1) / 5; ++i)
  {
    vector<int> my5;
    for(int j = 0; j < 5; ++j)
    {
      my5.push_back(a[p + 5 * i + j]);
    }
    sort(my5.begin(), my5.end());
    medians.push_back(my5[2]);
  }
  cout << "median is in:" << endl;
  disp(medians);
  int x = select(medians, 0, medians.size() - 1, (medians.size() -
    1) / 2 + 1);
  cout << "median is: " << x << endl << endl;

  vector<int> my_a(a);
  int i = my_partition(my_a, p, r, x);
  int j = i - p + 1;
  cout << "after partition, a becomes:" << endl;
  disp(my_a);
  cout << endl;

  if(k <= j)
  {
    vector<int> l_a(i - p + 1);
    copy(my_a.begin() + p, my_a.begin() + i + 1, l_a.begin());
    cout << "next, find in left part:" << endl;
    disp(l_a);
    cout << endl;

    cout << endl << "left select from left recursion" << cnt <<
```

```
    endl;
98      return select(l_a, 0, l_a.size() - 1, k);
99    }
100   else
101   {
102     vector<int> r_a(r - i);
103     copy(my_a.begin() + i + 1, my_a.begin() + r + 1, r_a.begin());
104     cout << "next, find in right part:" << endl;
105     disp(r_a);
106     cout << endl;
107
108     cout << endl << "left select from right recursion" << cnt <<
        endl;
109     return select(r_a, 0, r_a.size() - 1, k - j);
110   }
111 }
112
113 int main()
114 {
115   vector<int> a = {
116     29,22,28,14,45,
117     10,44,23, 9,39,
118     38,52, 6, 5,50,
119     37,11,26, 3,15,
120      2,53,40,54,25,
121     42,12,19,30,16,
122     18,13, 1,48,41,
123     24,43,46,47,17,
124     34,20,31,32,33,
125     35, 4,49,51, 7,
126     36,27, 8,21
127   };
128
129   cout << "a is:" << endl;
130   disp(a);
131   cout << endl;
132
133   cout << "the 15th least element of a is: " << select(a, 0, a.size
        () - 1, 15) << endl << endl;
134
135   cout << "sorted a is:" << endl;
136   sort(a.begin(), a.end());
137   disp(a);
138   cout << "the 15th least element of a is: " << a[14] << endl;
139
140   return 0;
141 }
```

```
1 a is:
2 29 22 28 14 45 10 44 23 9 39 38 52 6 5 50 37 11 26 3 15 2 53
      40 54 25 42 12 19 30 16 18 13 1 48 41 24 43 46 47 17 34
     20 31 32 33 35 4 49 51 7 36 27 8 21
3
4 call select 1:
```

```
find median of medians:
median is in:
28 23 38 15 40 19 18 43 32 35
call select 2:
find median of medians:
median is in:
28 32
call select 3:
recursion exit:
the sub-vector whose length is no greater than 5 is:
28 32
return: 28


left select from recursion exit3
median is: 28

after partition, a becomes:
18 23 19 15 28 40 38 43 32 35

next, find in left part:
18 23 19 15 28


left select from left recursion3
call select 4:
recursion exit:
the sub-vector whose length is no greater than 5 is:
18 23 19 15 28
return: 28


left select from recursion exit4
median is: 28

after partition, a becomes:
21 22 8 14 27 10 7 23 9 4 20 17 6 5 24 1 11 26 3 15 2 13 18
    16 25 19 12 28 30 42 54 40 53 48 41 37 43 46 47 50 34 52
    31 32 33 35 38 49 51 39 36 44 45 29

next, find in left part:
21 22 8 14 27 10 7 23 9 4 20 17 6 5 24 1 11 26 3 15 2 13 18
    16 25 19 12 28


left select from left recursion4
call select 5:
find median of medians:
median is in:
21 9 17 11 16
```

```
call select 6:
recursion exit:
the sub-vector whose length is no greater than 5 is:
21 9 17 11 16
return: 16


left select from recursion exit6
median is: 16

after partition, a becomes:
12 13 8 14 2 10 7 15 9 4 3 11 6 5 1 16 24 26 17 20 23 27 18
    21 25 19 22 28

next, find in left part:
12 13 8 14 2 10 7 15 9 4 3 11 6 5 1 16


left select from left recursion6
call select 7:
find median of medians:
median is in:
12 9 5
call select 8:
recursion exit:
the sub-vector whose length is no greater than 5 is:
12 9 5
return: 9


left select from recursion exit8
median is: 9

after partition, a becomes:
1 5 8 6 2 3 7 4 9 12 15 11 10 14 13 16

next, find in right part:
12 15 11 10 14 13 16


left select from right recursion8
call select 9:
find median of medians:
median is in:
12
call select 10:
recursion exit:
the sub-vector whose length is no greater than 5 is:
12
return: 12
```

```
101
102
103 left select from recursion exit10
104 median is: 12
105
106 after partition, a becomes:
107 10 11 12 15 14 13 16
108
109 next, find in right part:
110 15 14 13 16
111
112
113 left select from right recursion10
114 call select 11:
115 recursion exit:
116 the sub-vector whose length is no greater than 5 is:
117 15 14 13 16
118 return: 15
119
120
121 left select from recursion exit11
122 the 15th least element of a is: 15
123
124 sorted a is:
125 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
      24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42
      43 44 45 46 47 48 49 50 51 52 53 54
126 the 15th least element of a is: 15
```