注意在这里当q[u].clear()的时候，它map里的元素自己就没了

如果按照题意模拟那么需要deque<int> q[150000]

会爆内存MLE

用map！！！！现用现开，没用到的不会开内存

就这么就解决了

#include <cstdio>

#include <cstring>

#include <iostream>

#include<bits/stdc++.h>

using namespace std;

typedef long long ll;

void read(int &x)

{

char ch = getchar();

x = 0;

for (; ch < '0' || ch > '9'; ch = getchar());

for (; ch >='0' && ch <= '9'; ch = getchar()) x = x \* 10 + ch - '0';

}

map<int,deque<int> >q;

int main()

{

//freopen("input.txt","r",stdin);

int n,Q,t,u,v,w,val;

while(~scanf("%d%d",&n,&Q))

{

q.clear();

while(Q--)

{

read(t);

if(t==1)

{

read(u);

read(w);

read(val);

if(w==0)

q[u].push\_front(val);

else q[u].push\_back(val);

}

else if(t==2)

{

read(u);

read(w);

if(q[u].empty())

{

puts("-1");

continue;

}

if(w==0)

{

printf("%d\n",q[u].front());

q[u].pop\_front();

}

else

{

printf("%d\n",q[u].back());

q[u].pop\_back();

}

}

else if(t==3)

{

read(u);

read(v);

read(w);

if(w==0)

{

q[u].insert(q[u].end(),q[v].begin(),q[v].end());

}

else

{

q[u].insert(q[u].end(),q[v].rbegin(),q[v].rend());

}

q[v].clear();

}

}

}

return 0;

}

然而用list开150000却可以开的下

#include<iostream>

#include<map>

#include<deque>

#include<bits/stdc++.h>

using namespace std;

void read(int &x)

{

char ch = getchar();x = 0;

for (; ch < '0' || ch > '9'; ch = getchar());

for (; ch >='0' && ch <= '9'; ch = getchar()) x = x \* 10 + ch - '0';

}

list<int>\_list[150010];

int main()

{

int n,q,opt,u,val,w,v;

while(~scanf("%d%d",&n,&q))

{

for(int i=1;i<=n;i++)

\_list[i].clear();

while(q--)

{

read(opt);

if(opt==1)

{

read(u);read(w);read(val);

if(w==0)

\_list[u].push\_front(val);

else

\_list[u].push\_back(val);

}

else if(opt==2)

{

read(u);read(w);

if(\_list[u].empty())

{

puts("-1");

continue;

}

if(w==0)

{

printf("%d\n",\_list[u].front());

\_list[u].pop\_front();

}

else

{

printf("%d\n",\_list[u].back());

\_list[u].pop\_back();

}

}

else if(opt==3)

{

read(u);read(v);read(w);

if(w==0)

{

\_list[u].splice(\_list[u].end(),\_list[v]);

}

else

{

reverse(\_list[v].begin(),\_list[v].end());

\_list[u].splice(\_list[u].end(),\_list[v]);

}

\_list[v].clear();

}

}

}

return 0;

}

不过最好是这样，读入n之后再开list，list开到n+10

#include<iostream>

#include<map>

#include<deque>

#include<bits/stdc++.h>

using namespace std;

void read(int &x)

{

char ch = getchar();x = 0;

for (; ch < '0' || ch > '9'; ch = getchar());

for (; ch >='0' && ch <= '9'; ch = getchar()) x = x \* 10 + ch - '0';

}

int main()

{

int n,q,opt,u,val,w,v;

while(~scanf("%d%d",&n,&q))

{

list<int> \_list[n+10];

for(int i=1;i<=n;i++)

\_list[i].clear();

while(q--)

{

read(opt);

if(opt==1)

{

read(u);read(w);read(val);

if(w==0)

\_list[u].push\_front(val);

else

\_list[u].push\_back(val);

}

else if(opt==2)

{

read(u);read(w);

if(\_list[u].empty())

{

puts("-1");

continue;

}

if(w==0)

{

printf("%d\n",\_list[u].front());

\_list[u].pop\_front();

}

else

{

printf("%d\n",\_list[u].back());

\_list[u].pop\_back();

}

}

else if(opt==3)

{

read(u);read(v);read(w);

if(w==0)

{

\_list[u].splice(\_list[u].end(),\_list[v]);

}

else

{

reverse(\_list[v].begin(),\_list[v].end());

\_list[u].splice(\_list[u].end(),\_list[v]);

}

\_list[v].clear();

}

}

}

return 0;

}

更气人都是，将deque也定义在里面

deque<int> q[n+10];

就过了！！！！！

#include <cstdio>

#include <cstring>

#include <iostream>

#include<bits/stdc++.h>

using namespace std;

typedef long long ll;

void read(int &x)

{

char ch = getchar();

x = 0;

for (; ch < '0' || ch > '9'; ch = getchar());

for (; ch >='0' && ch <= '9'; ch = getchar()) x = x \* 10 + ch - '0';

}

deque<int>::iterator it;

int main()

{

//freopen("input.txt","r",stdin);

int n,Q,t,u,v,w,val;

while(~scanf("%d%d",&n,&Q))

{

deque<int> q[n+10];

while(Q--)

{

read(t);

if(t==1)

{

read(u);

read(w);

read(val);

if(w==0)

q[u].push\_front(val);

else q[u].push\_back(val);

}

else if(t==2)

{

read(u);

read(w);

if(q[u].empty()!=0)

{

puts("-1");

continue;

}

if(w==0)

{

printf("%d\n",q[u].front());

q[u].pop\_front();

}

else

{

printf("%d\n",q[u].back());

q[u].pop\_back();

}

}

else if(t==3)

{

read(u);

read(v);

read(w);

if(w==0)

{

while(q[v].empty()==0)

{

q[u].push\_back(q[v].front());

q[v].pop\_front();

}

}

else

{

while(q[v].empty()==0)

{

q[u].push\_back(q[v].back());

q[v].pop\_back();

}

}

}

}

}

return 0;

}