

University of Central Florida



Tyler Mark, X, X

```
1 Contest
                                                         1
2 Data structures
                                                         \mathbf{2}
3 Geometry
                                                         3
4 Graphs
                                                         4
5 Mathematics
                                                         5
6 Numerical Methods
                                                         6
7 Strings
                                                         7
Contest (1)
template.cpp
#include <bits/stdc++.h>
#define all(x) begin(x), end(x)
using namespace std;
\#define rep(i, a, b) for(int i = a; i < (b); ++i)
#define sz(x) (int) (x).size()
\#define all(x) begin(x), end(x)
using ll = long long;
using ld = long double;
using pii = pair<int, int>;
using vi = vector<int>;
using vii = vector<pii>;
using vvi = vector<vi>;
int main() {
 cin.tie(0)->sync_with_stdio(0);
```

cin.exceptions(cin.failbit);

return 0;

UCF	template	
$\underline{\text{Data structures}} \ (2)$		

UCF

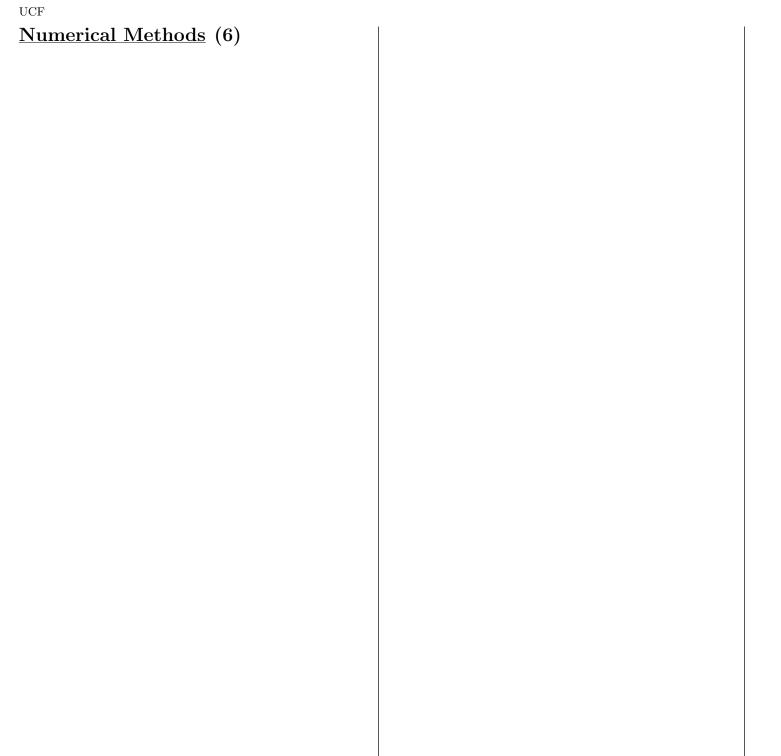
```
Geometry (3)
```

```
point.h
                                                     eafa48, 24 lines
 * Author: Tyler Marks
 * Description: Handles vector/point operations in the
      cartesian plane
template<class T> struct pnt {
   Тх, у;
   pnt(int _x = 0, int _y = 0): x(_x), y(_y) {}
    bool operator<(pnt o) { return pii(x, y) < pii(o.x, o.y); }</pre>
    bool operator==(pnt o) { return pii(x, y) == pii(o.x, o.y);
   pnt operator+(pnt o) { return pnt(x + o.x, y + o.y); }
   pnt operator-(pnt o) { return pnt(x - o.x, y - o.y); }
   pnt operator*(T c) { return pnt(x*c, y*c); }
   T dot(pnt o) { return x*o.x + y*o.y; }
   T cross(pnt o) { return x*o.y - y*o.x; }
   T cross(pnt o1, pnt o2) { return (o1-*this).cross(o2-*this)
   T dist2() { return dot(*this); }
    ld dist() { return sqrt(dist2()); }
   pnt unit() { return *this * (1/dist()); }
   pnt perp() { return pnt(-y, x); }
   pnt norm() { return perp().unit(); }
    ld ang() { return atan21(y, x); }
```

point 3

UCF Graphs (4)

UCF	
$\underline{\text{Mathematics}}$ (5)	
、	
	I



UCF $\underline{\text{Strings}}$ (7)