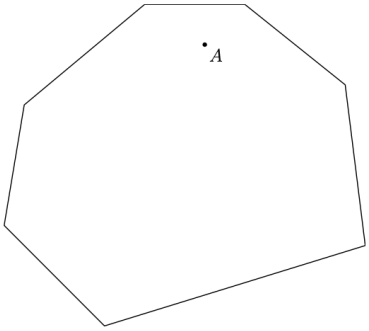
CS340 Analysis of Algorithms
Fall 2025

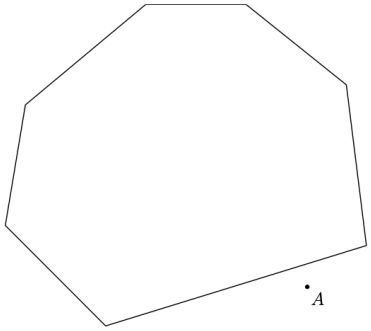
Lab:	**3**	**Professor:**	**Elizabeth Dinella**
Date:		**E-mail:**	`edinella@brynmawr.edu`
Title:	**Run Times**	**URL:**	**https://bmc-cs-340.github.io**

1 Point inside Polygon

Given a point $p=(p_x,p_y)$ and a convex polygon specified by n vertices $(x_1,y_1),(x_2,y_2),...,(x_n,y_n)$, design an algorithm to determine if p is inside or outside of the polygon (on the perimeter is considered inside).

1. Briefly describe your idea in English.
2. Give high-level pseudo code outlining your idea (no need to specify geometric computation details, assume that you can call functions).
3. Briefly argue your algorithm's correctness
4. State your algorithm's run time using Big-Oh
5. Show how your algorithm works on the following inputs





2 Estimating Run Times

For each of the following pseudo code segments, estimate their run time using the most appropriate Big-Oh expression of n. Assume that the lines "a constant number of steps" do not modify any of hte loop control variables.

Function `A`(*input: n elements*)
| `a constant number of steps`

Function `B`(*input: n elements*)
| **for** i=1 `to` $\lfloor\sqrt{n}\rfloor$ **do**
| | `a constant number of steps`
| **end**
| **for** j=1 `to` $\lfloor\sqrt{n}\rfloor$ **do**
| | `a constant number of steps`
| **end**

Function `C`(*input: n elements*)
| **for** i=1 `to` $n/2$ **do**
| | **for** j=1 `to` $n/2$ **do**
| | | `a constant number of steps`
| | **end**
| **end**

Function `D`(*input: n elements*)
| **for** i=1 `to` $\sqrt{n}$ **do**
| | **for** j=1 `to` $\sqrt{n}$ **do**
| | | `a constant number of steps`
| | **end**
| **end**

Function `E`(*input: n elements*)
| i=1
| **while** $i \leq n$ **do**
| | **for** j=1 `to` n **do**
| | | `a constant number of steps`
| | **end**
| | i=$i\times 2$
| **end**

```
Function F(input: n elements)
    let P=1,2,...,n
    let S be the set of all permutations of P of size 3
    for each s∈S do
        a constant number of steps
    end
```

```
Function G(input: n elements)
    let P=1,2,...,n
    let S be the set of all permutations of P
    for each s∈S do
        a constant number of steps
        print s
    end
```

```
Function H(input: n elements)
    i=1
    j=1
    while i≤n do
        while j≤n do
            a constant number of steps
            i++
            j++
        end
    end
```