In [10]:

```python
import matplotlib.pyplot as plt
```

In [2]:

```python
def orientationTest(p1,p2,p3):
    """
    tests the orientation of p1-p2-p3 path by calculating
    the determinant of x1 y1 1
                       x2 y2 1
                       x3 y3 1

       "U turn" is considered to be straight turn
    Parameters
    ----------
    p1,p2,p3: tuple of point coordinates (x,y) dtype = float

    Return
    ------
    direction: str
    """
    x1,y1 = p1
    x2,y2 = p2
    x3,y3 = p3
    determinant = x1*y2 - x2*y1 + x2*y3 -x3*y2 + x3*y1 - x1*y3

    if determinant == 0:
        return 'straight'
    elif determinant > 0:
        return 'left'
    else:
        return 'right'
```

In [5]:

```python
# test cases
p1 = (0,0)
p2 = (0,1)
p3 = (0,3)
assert(orientationTest(p1,p2,p3) == 'straight')
p2 = (1,1)
assert(orientationTest(p1,p2,p3) == 'left')
p2 = (-1,1)
assert(orientationTest(p1,p2,p3) == 'right')
p2 = p3 = p1
assert(orientationTest(p1,p2,p3) == 'straight')
```

In [37]:

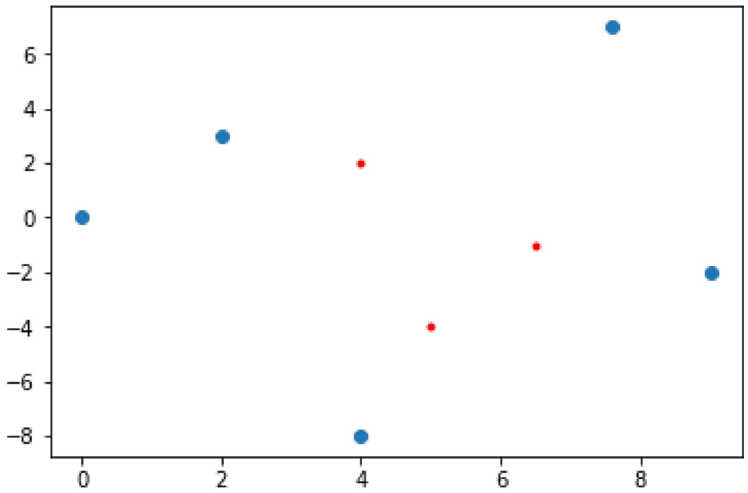
```
def convex_hull(pts):
    """
    find convex_hull of a set of points
    sort(n) + O(n) -> O(nlogn) for build-in Timesort

    Parameter
    ---------
    pts: array of tuples (dtype=(float,float))
        x,y coordinates of the points

    Return
    ------
    convexHull: set of tuples (dtype=(float,float))
        x,y coordianates of the points on the convexHull
    """
    stack = []
    sorted_pts = sorted(pts)
    # get upper hull
    for pt in sorted_pts:
        while len(stack) >= 2 and orientationTest(stack[-2],stack[-1],pt) != 'right':
            stack.pop()
        stack.append(pt)
    hull = set(stack)
    stack = []
    # get lower hull
    for pt in sorted_pts:
        while len(stack) >= 2 and orientationTest(stack[-2],stack[-1],pt) != 'left':
            stack.pop()
        stack.append(pt)

    hull |= set(stack)
    return hull
```

In [45]:

```
# test cases
pts = [(0,0),(2,3),(4,2),(6.5,-1),(7.6,7),(9,-2),(4,-8),(5,-4)]
x,y = zip(*pts)
plt.plot(x,y,'r.')
convex_hull_pts = convex_hull(pts)
x_c,y_c = zip(*convex_hull_pts)
plt.plot(x_c,y_c, 'o')
```

Out[45]:

```
[<matplotlib.lines.Line2D at 0x19af2948898>]
```


6
4
2
0
−2
−4
−6
−8
0
2
4
6
8