

题目一

1. 算法伪代码

```
function reverse_symmetric_matvec(a, v):  
    n = length(v)  
    result = [0 for i in 0..n-1]  
    for i = 0 to n-1:  
        sum = 0  
        for j = 0 to n-1:  
            sum += a[n-1 + i - j] * v[j]  
        result[i] = sum  
    return result
```

2. 时间复杂度

外层循环运行 n 次，内层循环也运行 n 次，每次都执行常数操作。

整体复杂度为： $O(n^2)$ 。

题目二

1. 算法伪代码

```
def wildcard_fft_match(S, P):  
    n, m = len(S), len(P)  
    N = 1 << (n + m - 1).bit_length() # padding to next power of 2  
    chars = set(S + P)  
    chars.discard('?')  
  
    total_score = np.zeros(N)  
    wildcard_count = sum(1 for ch in P if ch == '?')  
  
    for c in chars:  
        Sc = np.array([1 if ch == c else 0 for ch in S] + [0] * (N - n))  
        Pc = np.array([1 if ch == c else 0 for ch in P[:-1]] + [0] * (N - m))  
        conv = np.real(iff(fft(Sc) * fft(Pc)))  
        total_score += conv  
  
    result = []
```

```
threshold = len(P) - wildcard_count
for i in range(n - m + 1):
    if int(round(total_score[i + m - 1])) == threshold:
        result.append(i)
return result
```