



人工智能的硬件基石：从物理器件到计算架构

2025年春季 - 作业4

1、缓存微架构 (40分)

考虑以下访存模式：A, B, A, C, B, A。假设A、B和C是分别位于不同缓存块 (different cache block) 的地址。此外，假设A、B和C是以均匀随机的方式生成的，并且使用LRU (最近最少使用) 替换机制。若两个缓存块 (cache block) 同时为无效 (invalid)，则任何给定的访存块都有相等的机会放置在任意的Cache Way中。请提供简单的一两句说明。

如果出现以下情况，第3次访问“A”命中 (Hit) 的概率是多少？

- 1) 缓存有2行，并且是fully-associative的。 [10分]
- 2) 缓存有4行，并且是fully-associative的。 [10分]
- 3) 缓存有8行，并且是direct-mapped的。 [10分]
- 4) 缓存有4行，并且是2-way set associative的。 [10分]

2、缓存局部性与矩阵计算 (60分)

假设我们要完成 $C = AB$ 的矩阵乘法，矩阵的大小都为 $n \times n$ ，处理器有一个3行全相连的缓存，每行可以存储 $n/2$ 个内存中连续的元素值。请比较以下两种代码方式下，要完成矩阵运算所需要的内存访问次数；如果将每行缓存扩大为存储 n 个内存中连续的元素值，重新两种方式所需的访存次数。假设A, B, C矩阵在内存中都按行连续存储 (以A矩阵为例，内存中的存储顺序为 $a[0][0], a[0][1], \dots, a[0][n-1], a[1][0], \dots, a[1][n-1], \dots$)。

```
(1)   for (int i = 0; i < n; i++)
        for (int j = 0; j < n; j++)
            for (int k = 0; k < n; k++)
                c[i][j] += a[i][k] * b[k][j];
```

```
(2)    for (int i = 0; i < n; i++)  
        for (int k = 0; k < n; k++)  
            for (int j = 0; j < n; j++)  
                c[i][j] += a[i][k] * b[k][j];
```