Solving Problems using Dynamic Programming

$$\therefore \mathbf{f_n}(\mathbf{k},\ \mathbf{p}) = \begin{cases} \mathbf{1} & \text{if } \mathbf{k} \equiv \mathbf{0} \text{ and } \mathbf{p} \equiv \mathbf{0} \\ \sum\limits_{\delta \in [\mathbf{1},\ \beta]} \{\mathbf{f_{n-1}}(\mathbf{k}-\mathbf{1},\ \mathbf{p}-\delta)\} & \text{otherwise} \end{cases}$$

A Hackers' Perspective

π^{∞}

```
function perfectkriya($\alpha$)
    $f[0..\alpha] \leftarrow \{\infty\}$
    $f[0] \leftarrow 0$
    for $\beta \in [1,\ \alpha]$ do
        for $\gamma \in [1,\ \sqrt{\beta}]$ do
            $f[\beta] \leftarrow \min\left(f[\beta],\ f[\beta - \gamma^2] + 1\right)$
        end for
    end for
    return $f[\alpha]$
end function
```

```
int firstkriya(int beta, int alpha)
{
    // max no of Kriyas with beta Pranayams
    int n = std::min(beta, alpha);
    std::vector<int> f(n, 0);
    f(0) = 1;
    for(int p = 1; p <= beta; p++)
    {
        int prev = 0, cur = 0;
        for(int k = 0; k < n; k++)
        {
            cur = f(k);
            f(k) += prev + (k+1 < n ? f(k+1) : 0);
            prev = cur;
        }
    }
    return f(0);
}
```

Chandra Shekhar Kumar

Ancient Science Publishers