

数理・計算科学特論 C 第 4 回

河瀬 康志

2021 年 6 月 22 日

- ① スキーレンタル問題
- ② Yao の原理
- ③ ページング問題
- ④ まとめ

スキーレンタル問題（再）

これからスキーを始めるが、今後何回スキーに行くかわからない。
スキー板を買うべきか、レンタルすべきか？

- レンタルすると、1 回につき 1 万円
 - 買うと N 万円
 - 買ったスキーは、ずっと使い続けることができる（壊れない）
-
- 最適なオンラインアルゴリズムは $N - 1$ 回のレンタル後に購入
 - 最悪の状況は N 回スキーに行く場合で、競合比は $\frac{(N-1)+N}{N} = 2 - \frac{1}{N}$

スキーレンタル問題 (再)

これからスキーを始めるが、今後何回スキーに行くかわからない。
スキー板を買うべきか、レンタルすべきか？

- レンタルすると、1 回につき 1 万円
 - 買うと N 万円
 - 買ったスキーは、ずっと使い続けることができる (壊れない)
-
- 最適なオンラインアルゴリズムは $N - 1$ 回のレンタル後に購入
 - 最悪の状況は N 回スキーに行く場合で、競合比は $\frac{(N-1)+N}{N} = 2 - \frac{1}{N}$
- 最悪の状況を考えるのは悲観的すぎる
- 乱数を使うことで最悪を回避する (今日)
 - 入力 of 分布を考える (来週)

乱択アルゴリズムに対する競合比

$$\sup_{\text{入力列}} \frac{\mathbb{E}[\text{乱択オンラインアルゴリズムでのコスト}]}{\text{入力列を知っていた場合の最適コスト}}$$

- 期待値の意味で最適コストの何倍以下に抑えられるかを保証
- 乱択アルゴリズムに対して最悪の入力列を考える (乱数の値には依存しない)
- 漸近的競合比も同様に定義できる (この授業では扱わない)
- 別の定義の仕方もある (今日の後半)

確率的方法による改善

アルゴリズム： k 回目に確率 $p_k := \gamma \cdot \delta^{k-1}$ で購入 ($k = 1, \dots, N$)

$$\sum_{i=1}^N p_k = 1$$

- スキーに k ($\leq N$) 回行く時

- ▶ 最適コスト = k

- ▶ アルゴリズムの期待コスト = $\sum_{i=1}^k [p_i \cdot N + (1 - \sum_{j=1}^i p_j)] = \sum_{i=1}^k \frac{1}{1-\delta^{-N}}$

→ 競合比は $\frac{1}{1-\delta^{-N}} = \frac{1}{1-(1-1/N)^N} \leq \frac{e}{e-1} \approx 1.582$

- スキーに N 回以上行く時

- ▶ 最適コスト = N

- ▶ アルゴリズムの期待コスト = $\sum_{i=1}^N \frac{1}{1-\delta^{-N}}$

→ 競合比は $\frac{1}{1-\delta^{-N}} = \frac{1}{1-(1-1/N)^N} \leq \frac{e}{e-1} \approx 1.582$

確率的方法による改善

アルゴリズム： k 回目に確率 $p_k := \gamma \cdot \delta^{k-1}$ で購入 ($k = 1, \dots, N$)

$$\sum_{i=1}^N p_k = 1$$

- スキーに k ($\leq N$) 回行く時

- ▶ 最適コスト = k

- ▶ アルゴリズムの期待コスト = $\sum_{i=1}^k [p_i \cdot N + (1 - \sum_{j=1}^i p_j)] = \sum_{i=1}^k \frac{1}{1 - \delta^{-N}}$

→ 競合比は $\frac{1}{1 - \delta^{-N}} = \frac{1}{1 - (1-1/N)^N} \leq \frac{e}{e-1} \approx 1.582$

- スキーに N 回以上行く時

- ▶ 最適コスト = N

- ▶ アルゴリズムの期待コスト = $\sum_{i=1}^N \frac{1}{1 - \delta^{-N}}$

→ 競合比は $\frac{1}{1 - \delta^{-N}} = \frac{1}{1 - (1-1/N)^N} \leq \frac{e}{e-1} \approx 1.582$

下界はどう示す？ 全ての乱択アルゴリズムを考えるのは大変

アウトライン

- 1 スキーレンタル問題
- 2 Yao の原理
- 3 ページング問題
- 4 まとめ

乱択アルゴリズムの表現

乱択アルゴリズムは決定性アルゴリズム上の確率分布として表現できる

- スキーレンタル問題に対する乱択アルゴリズムでは、
何回目に購入するかを最初に決定 (k 回目に購入する確率 p_k)
→ k 回目に購入という決定性アルゴリズムを確率 p_k で選択している
- しかし、一般には入力に依存してランダムな挙動を取るかもしれない
例: ページング問題においてページフォルトが起きたらランダムに 1 つ消去するなど
- このような場合も、どのような乱数 r を途中で利用するかを引数とするアルゴリズムと考えることで、決定的なアルゴリズム $\text{ALG}(\sigma, r)$ に変換できる

ゲーム理論的解釈

我々 vs. 敵

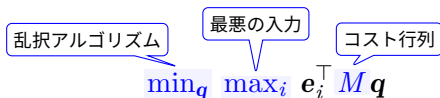
最悪ケース解析は二人零和ゲームだと解釈可能

- 我々の戦略は決定性アルゴリズム 混合戦略が乱択アルゴリズム
- 敵の戦略は入力
- 2人の戦略が決まるとコストが決定される コストは目的関数値, 競合比, 計算時間, など

例: スキーレンタル問題 ($N = 5$) のコスト行列 (コストは目的関数値)

敵 \ 我々	1	2	3	4	5	6	...
1	5	1	1	1	1	1	...
2	5	6	2	2	2	2	...
3	5	6	7	3	3	3	...
4	5	6	7	8	4	4	...
5	5	6	7	8	9	5	...
6	5	6	7	8	9	10	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	

最悪ケース解析



乱択アルゴリズム

最悪の入力

コスト行列

$$\min_q \max_i e_i^\top M q$$

- 最悪の入力に対してコストの小さい乱択アルゴリズムを設計したい
- 乱択アルゴリズムを選択した後に敵が最悪の入力を選択する
- 本当は $\inf_q \sup_i e_i M q$ を考える必要がある

ミニマックス定理

後手有利

$$\max_p \min_q \mathbf{p}^\top M \mathbf{q} \leq \min_q \max_p \mathbf{p}^\top M \mathbf{q}$$

von Neumann のミニマックス定理

$$\max_p \min_q \mathbf{p}^\top M \mathbf{q} = \min_q \max_p \mathbf{p}^\top M \mathbf{q}$$

どちらが先手でも得られる辿り着く値は同じ

Loomis の補題

$$\max_p \min_j \mathbf{p}^\top M \mathbf{e}_j = \min_q \max_i \mathbf{e}_i^\top M \mathbf{q}$$

後手は最適戦略として純粋戦略を取れる

- M : $n \times m$ 行列 von Neumann の定理の成立には有限である必要あり
- $\mathbf{p}$: $\{1, 2, \dots, n\}$ 上の確率分布 (混合戦略)
- $\mathbf{q}$: $\{1, 2, \dots, m\}$ 上の確率分布 (混合戦略)

Yao の原理 (基本形)

Loomis の補題

$$\max_{\mathbf{p}} \min_j \mathbf{p}^\top M \mathbf{e}_j = \min_{\mathbf{q}} \max_i \mathbf{e}_i^\top M \mathbf{q}$$

$\mathbf{p}$ として具体的な確率分布を 1 つもってくることで、
ゲームの値の下界を見積もることができる

Yao の原理 (基本形)

任意の確率分布 $\hat{\mathbf{p}}$ について

$$\min_j \hat{\mathbf{p}}^\top M \mathbf{e}_j \leq \min_{\mathbf{q}} \max_i \mathbf{e}_i^\top M \mathbf{q}$$

例: 2 人零和ゲームの例

最大化したい	最小化したい				
	A	左	中	右	
B		上	40	10	30
		下	0	20	10

$$V = \min_{\mathbf{q}} \max_i \mathbf{e}_i^\top M \mathbf{q}$$

- A が左を確率 0.2, 中を 0.8 で選ぶ
 - B が上と下のどちらを選んでも値は 16
 - V は 16 以下
- B が上を確率 0.4, 下を 0.6 で選ぶ
 - A が左か中を選べば値は 16. A が右を選べば値は 18
 - V は 16 以上 (Yao の原理)

$$\min_{\text{乱択アルゴリズム}} \max_{\text{入力列}} \frac{\mathbb{E}[\text{乱択アルゴリズムでのコスト}]}{\text{入力列を知っていた場合の最適コスト}}$$

- 式で書き直すと $\min_q \max_{\sigma} \frac{\mathbb{E}_{y \sim q}[\text{ALG}_y(\sigma)]}{\text{OPT}(\sigma)}$ を評価したい
- Yao の原理を用いると、入力分布 $\hat{p}$ に対し、次の下界を得る

$$\min_{\text{ALG}} \mathbb{E}_{x \sim \hat{p}} \left[\frac{\text{ALG}(\sigma_x)}{\text{OPT}(\sigma_x)} \right]$$

- しかし、これは使い勝手があまりよくない (最適なアルゴリズムを見つけるのは難しい)

Yao の原理 (競合比)

最適な決定性アルゴリズムでの期待コスト

期待最適コスト

ある入力列上の分布 $\hat{p}$ に対し $\min_{y \in \mathcal{Y}} \mathbb{E}_{\hat{p}}[\text{ALG}_y(\sigma_x)] \geq c \cdot \mathbb{E}_{\hat{p}}[\text{OPT}(\sigma_x)]$

$\implies$ 任意の乱択オンラインアルゴリズムの競合比は c 以上

- $\{\sigma_x \mid x \in \mathcal{X}\}$: 入力列の集合
- $\{\text{ALG}_y \mid y \in \mathcal{Y}\}$: 決定性アルゴリズムの集合

乱択アルゴリズムに対する競合比は「競合比の期待値」ではないので一手間必要

証明 ($\mathcal{X}, \mathcal{Y}$ が有限のとき)

仮定

Yao の原理 (基本形)

- $0 \leq \min_{y \in \mathcal{Y}} \mathbb{E}_{\hat{p}}[\text{ALG}_y(\sigma_x) - c \cdot \text{OPT}(\sigma_x)] \leq \min_q \max_{x \in \mathcal{X}} \mathbb{E}_q[\text{ALG}_y(\sigma_x) - c \cdot \text{OPT}(\sigma_x)]$
- 各 q に対し, $\max$ を達成する $x \in \mathcal{X}$ を $x(q)$ と書くと $\frac{\mathbb{E}_q[\text{ALG}_y(\sigma_{x(q)})]}{\text{OPT}(\sigma_{x(q)})} \geq c$

Yao の原理 (競合比)

最適な決定性アルゴリズムでの期待コスト

期待最適コスト

ある入力列上の分布 $\hat{p}$ に対し $\min_{y \in \mathcal{Y}} \mathbb{E}_{\hat{p}}[\text{ALG}_y(\sigma_x)] \geq c \cdot \mathbb{E}_{\hat{p}}[\text{OPT}(\sigma_x)]$

$\implies$ 任意の乱択オンラインアルゴリズムの競合比は c 以上

- $\{\sigma_x \mid x \in \mathcal{X}\}$: 入力列の集合
- $\{\text{ALG}_y \mid y \in \mathcal{Y}\}$: 決定性アルゴリズムの集合

乱択アルゴリズムに対する競合比は「競合比の期待値」ではないので一手間必要

証明 ($\mathcal{X}, \mathcal{Y}$ が有限のとき)

仮定

Yao の原理 (基本形)

- $0 \leq \min_{y \in \mathcal{Y}} \mathbb{E}_{\hat{p}}[\text{ALG}_y(\sigma_x) - c \cdot \text{OPT}(\sigma_x)] \leq \min_q \max_{x \in \mathcal{X}} \mathbb{E}_q[\text{ALG}_y(\sigma_x) - c \cdot \text{OPT}(\sigma_x)]$
- 各 q に対し, $\max$ を達成する $x \in \mathcal{X}$ を $x(q)$ と書くと $\frac{\mathbb{E}_q[\text{ALG}_y(\sigma_{x(q)})]}{\text{OPT}(\sigma_{x(q)})} \geq c$

$\mathcal{X}$ や $\mathcal{Y}$ が有限集合でない場合も同様の結果を示すことができる (証明略)

$\min_{y \in \mathcal{Y}}$ を $\inf_{y \in \mathcal{Y}}$ に変更する必要がある. ミニマックス定理は成り立たないが, $\inf \sup \geq \sup \inf$ が成り立つ

スキューラル問題の乱択競合比下界

入力の分布：ちょうど k 回行く確率が $\frac{1}{N} \cdot \left(1 - \frac{1}{N}\right)^{k-1}$

$$\sum_{k=1}^{\infty} \frac{1}{N} \cdot \left(1 - \frac{1}{N}\right)^{k-1} = 1$$

- 最適コストの期待値:

$$\sum_{k=1}^N \underbrace{\frac{k}{N} \cdot \left(1 - \frac{1}{N}\right)^{k-1}}_{N - 2N(1 - 1/N)^N} + \sum_{k=N+1}^{\infty} \underbrace{\frac{N}{N} \cdot \left(1 - \frac{1}{N}\right)^{k-1}}_{N(1 - 1/N)^N} = N \left(1 - \left(1 - \frac{1}{N}\right)^N\right)$$

- i 回目に行くときに購入するアルゴリズムの期待コスト:

$$\sum_{k=1}^{i-1} \underbrace{\frac{k}{N} \cdot \left(1 - \frac{1}{N}\right)^{k-1}}_{N - (1 - 1/N)^{i-1}(i + N - 1)} + \sum_{k=i}^{\infty} \underbrace{\frac{i-1+N}{N} \cdot \left(1 - \frac{1}{N}\right)^{k-1}}_{(1 - 1/N)^{i-1}(i + N - 1)} = N$$

Yao の原理より以下が成立

定理

任意の乱択オンラインアルゴリズムの競合比は $\frac{1}{1 - (1 - 1/N)^N}$ 以上

最初に紹介した乱択アルゴリズムの最適性を意味している

注意：最大化問題に対する競合比

$$\min_{\text{乱択アルゴリズム}} \max_{\text{入力列}} \frac{\text{入力列を知っていた場合の最適利得}}{\mathbb{E}[\text{乱択アルゴリズムでの利得}]}$$

- 最大化問題に対する乱択アルゴリズムの競合比は通常上のように定義
- 最小化問題の場合と違って、競合比の期待値とはならない
 - 入力の分布 $\hat{p}$ に対し $\min_{\text{ALG}} \mathbb{E}_{x \sim \hat{p}} \left[\frac{\text{OPT}(\sigma_x)}{\text{ALG}(\sigma_x)} \right]$ は上界とは限らない
- 何なら下界となる？（演習）

アウトライン

- 1 スキーレンタル問題
- 2 Yao の原理
- 3 ページング問題**
- 4 まとめ

ページング問題（再）

- 記憶装置は 2 段階
 - ▶ 低速記憶装置には N 個のページが蓄えられている
 - ▶ 高速記憶装置には k 個のページを蓄えることができる ($k < N$)
- 必要データを含むページを要求するページ要求が逐次的に与えられる
- 要求のあったページが高速記憶装置に入っていない場合は、低速記憶装置から高速記憶装置にデータをロードする必要がある（ページフォルト）
- ページフォルトの数をなるべく小さくしたい
- 決定性アルゴリズムでの競合比の下界は k
- FIFO や LRU を使えば競合比 k を達成

乱択アルゴリズムでの競合比の下界

入力の分布: $k + 1$ 種類のページを一樣ランダムに T 回要求

- オンラインアルゴリズムでのコスト
 - ▶ 各要求に対し確率 $\frac{1}{k+1}$ でフォルト
 - ▶ ページフォルトの回数の期待値は $\frac{T}{k+1}$
- 最適オフラインアルゴリズムでのコスト
 - ▶ $k + 1$ 種類の要求が揃う毎に 1 回フォルト (クーポンコレクター問題)
揃うまでの回数の期待値は $\sum_{i=1}^{k+1} \frac{k+1}{i} = (k+1) \log(k+1) = \Theta(k \log k)$
 - ▶ ページフォルトの回数の期待値は $\Theta(T/k \log k)$

elementary renewal theorem より成立. T を最適オフラインアルゴリズムが 2 回目のフォルトを起こすときとしても良い

Yao の原理より以下が成立

定理

任意の乱択オンラインアルゴリズムの競合比は $\Omega(\log k)$

MARKING アルゴリズム

高速記憶装置の全てのページをマーク;

while i 番目のページ要求 p_i が来る **do**

if p_i が高速記憶装置にない **then**

if 高速記憶装置にマークされていないページがある **then**

 マークされていないものからランダムに1つ消去する;

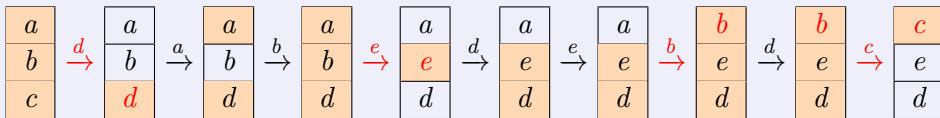
else マークをすべて消去し、ランダムに1つ消去する;

p_i を高速記憶装置に入れる

p_i にマークをつける

例

- 高速記憶装置 ($k = 3$) の初期データ: $\{a, b, c\}$
- ページ要求列: $d, a, b, e, d, e, b, d, c$

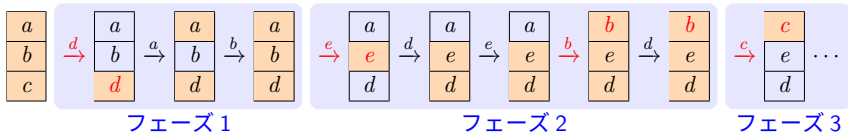


ページフォルト: 4 回

フェーズ

MARKING でのマーク消去のタイミングでの分割に一致

- ページ要求列を k 種類の要求からなる最長部分列（フェーズ）に分解



- 各フェーズ j について解析
 - S_j : フェーズ j 開始時に MARKING が保持しているページ集合
 - R_j : フェーズ j に要求されたページ集合
 - $R_j \setminus S_j$: New ページ, $|R_j \setminus S_j| = \ell_j$
 - $R_j \cap S_j$: Old ページ, $|R_j \cap S_j| = k - \ell_j$

MARKING アルゴリズムの競合比

定理

$$\text{MARKING の期待コスト} \leq 2 \sum_{i=1}^k 1/i \cdot H_k \cdot \text{最適コスト}$$

以下の 2 つの補題に分けて証明する

補題 1

フェーズ j での New ページの数

$$\text{MARKING の期待コスト} \leq H_k \cdot \sum_j \ell_j$$

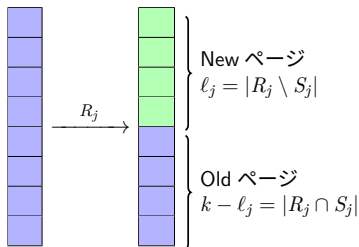
補題 2

$$\text{最適コスト} \geq \sum_j \ell_j / 2$$

補題 1 の証明

補題 1

$$\text{MARKING の期待コスト} \leq H_k \cdot \sum_j \ell_j$$

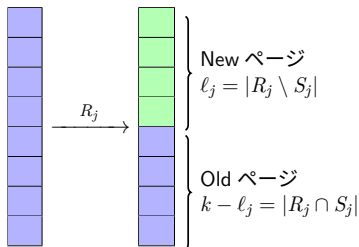


- 各 New ページについて 1 回フォルト
- フェーズ j において i 番目に要求された Old ページでのフォルト確率 $\leq \frac{\ell_j}{k - (i-1)}$
($i-1$) 個の Old ページはマーク済み.
残り $k - (i-1)$ 個のうち消去された Old ページは高々 ℓ_j 個

補題 1 の証明

補題 1

$$\text{MARKING の期待コスト} \leq H_k \cdot \sum_j \ell_j$$



- 各 New ページについて 1 回フォルト
- フェーズ j において i 番目に要求された Old ページでのフォルト確率 $\leq \frac{\ell_j}{k - (i-1)}$
($i-1$) 個の Old ページはマーク済み。
残り $k - (i-1)$ 個のうち消去された Old ページは高々 ℓ_j 個

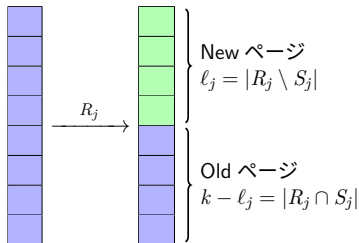
MARKING の期待コスト

$$\begin{aligned} &\leq \left(\frac{\ell_j}{k} + \frac{\ell_j}{k-1} + \cdots + \frac{\ell_j}{\ell_j+1} \right) + \ell_j \\ &\leq \left(\frac{\ell_j}{k} + \frac{\ell_j}{k-1} + \cdots + \frac{\ell_j}{\ell_j+1} \right) + \left(\frac{\ell_j}{\ell_j} + \frac{\ell_j}{\ell_j-1} + \cdots + \frac{\ell_j}{1} \right) = \ell_j \cdot H_k \end{aligned}$$

補題 2 の証明

補題 2

$$\text{最適コスト} \geq \sum_j \ell_j / 2$$



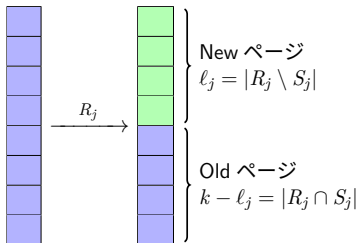
- フェーズ $j-1, j$ で計 $k + \ell_j$ 種のページ
- どんなアルゴリズムでも,
フェーズ $j-1, j$ で ℓ_j 回フォルト
- 2通りのペアリングを考えて解析

$$\{\underbrace{(0, 1)}_{\ell_1}, \underbrace{(2, 3)}_{\ell_3}, \dots\}, \{\underbrace{(1, 2)}_{\ell_2}, \underbrace{(3, 4)}_{\ell_4}, \dots\}$$

補題 2 の証明

補題 2

$$\text{最適コスト} \geq \sum_j \ell_j / 2$$



- フェーズ $j-1, j$ で計 $k + \ell_j$ 種のページ
- どんなアルゴリズムでも,
フェーズ $j-1, j$ で ℓ_j 回フォルト
- 2通りのペアリングを考えて解析

$$\{\underbrace{(0, 1)}_{\ell_1}, \underbrace{(2, 3)}_{\ell_3}, \dots\}, \{\underbrace{(1, 2)}_{\ell_2}, \underbrace{(3, 4)}_{\ell_4}, \dots\}$$

$$\text{最適コスト} \geq \max \left\{ \sum_j \ell_{2j-1}, \sum_j \ell_{2j} \right\} \geq \frac{1}{2} \left(\sum_j \ell_{2j-1} + \sum_j \ell_{2j} \right) = \sum_j \ell_j / 2$$

- 任意の決定性アルゴリズムの競合比は k 以上
- FIFO, LRU アルゴリズムの競合比は k

- 任意の乱択アルゴリズムの競合比は $\Omega(\log k)$

最初の要求を初期状態にないページとし, 以後は直前と異なる k 種類から一様ランダムに要求とすれば競合比の下界 H_k を示せる

- MARKING アルゴリズムの競合比は $2H_k = O(\log k)$ 以下
競合比 H_k のアルゴリズムも知られている [McGeopch and Sleator, 1991]

- ① スキーレンタル問題
- ② Yao の原理
- ③ ページング問題
- ④ まとめ

- 乱択アルゴリズムに対する競合比を定義
- 競合比の下界を求めるテクニックとして Yao の不等式を紹介
- スキーレンタル問題の乱択競合比は $\frac{1}{1-(1-1/N)^N} \approx \frac{e}{e-1}$
- ページング問題の乱択競合比は $H_k = \Theta(\log k)$

演習

- 最大化問題の競合比に対する Yao の原理はどうなるか？

クーポンコレクター問題

- n 種類のクーポンを揃えたい
 - 1 回の試行で n 種類のうち 1 つが当確率 $(1/n)$ でもらえる
 - 全部揃えるまでに必要な試行回数の期待値は何回か？
-
- X_k : k 種揃った状態から $k + 1$ 種類目を手に入れるまでにかかる回数
 - k 種揃った状態において新しいクーポンが手に入る確率は $\frac{n-k}{n}$ なので、 X_k は $p = \frac{n-k}{n}$ の幾何分布 ($P(i) = p(1-p)^{i-1}$)
 - 幾何分布の期待値は $1/p$ なので、 $\mathbb{E}[X_k] = \frac{n}{n-k}$
$$\sum_{i=1}^{\infty} i \cdot p(1-p)^{i-1} = 1/p$$
 - 必要な試行回数は $\mathbb{E}[\sum_{k=0}^{n-1} X_k] = \sum_{k=0}^{n-1} \mathbb{E}[X_k] = \sum_{k=1}^{n-1} \frac{n}{n-k} = n \cdot H_n$