Мы хотим написать парсер (а как..), который будет прогонять тикеры насдака с инвестинга и выбирать наиболее подходящие 20 самых рискованных, 20 нейтральных и 20 нерискованных акций.

1. Программа прогружает все тикеры насдака. По очереди вставляет колонкой дневные доходности в датафрейм
2. Программа подбирает все возможные комбинации портфеля по 20 акций. **Хеш таблица?**
3. Необходимо по каждому типу рискованности задать критерии ожидаемой доходности, волатильности итд (об этом далее под буквами), чтобы программа поняла, к какой категории относится каждый порфель. Результат по каждому критерию одним числом вписывать строкой в датафрейм, потом его отсортировать по колонкам
   1. Conservative – minimum volatility criterium – выдаст вариант с минимальной волатильностью
   2. Neutral – efficient risk, какое целевое значение ретерна выбрать? Как минимум выше ожидаемой инфляции + сколько %? – выдаст вариант с доходностью не менее заданной при минимальной волатильности. Здесь можно визуализировать данные на scatter plot ретерн (х)-волатильность(у), чтобы определиться, какая волатильность приемлема(?)
   3. Aggressive – Efficient return criterium (сколько волатильность?)
4. 20 верхних акций в каждой колонке датафрейма – то, что нам нужно
5. Вычисления доходностей каждого портфеля + сравнение с реальными данными

**Идея**: вместо датафрейма необходимое значение критерия записывать в хеш таблицу + записывать индекс тикера в датафрейме, чтобы не нужно было сортировать датафрейм 100 тысяч лет. Collision resolution – closed addressing.

**Проблема**: как писать хеш таблицу. Как выбрать и прописать hash function

*vol = σ√T*

* v = volatility over some interval of time
* σ =standard deviation of returns
* T = number of periods in the time horizon

*minimum volatility: Var(P)-> min*

*efficient risk: vol(P)-> min, how to maximize return here*

*efficient return: -> max, vol = ? does not really matter?*

*Примечание: нам нравится наша идея. Но на моменте «нам надо написать парсер» можно заканчивать*