In [ ]:
```
from thinkdsp import *
import numpy as np
import matplotlib.pyplot as plt
```

Лабораторная работа 6

Упражнение 6.1

> В этой главе утверждается, что `analyze1` требует времени пропорционально n^3, а `analyze2` - пропорционально n^2. Убедитесь в этом, запуская их с несколькими разными массивами и засекая время работы. Используйте команду %timeit

In [ ]:
```
from thinkdsp import UncorrelatedGaussianNoise

signal = UncorrelatedGaussianNoise()
noise = signal.make_wave(duration=1.0, framerate=16384)
```

Начнем с генерации сигнала белого шума

In [ ]:
```
PI2 = np.pi * 2

def analyze1(ys, fs, ts):
    args = np.outer(ts, fs)
    M = np.cos(PI2 * args)
    amps = np.linalg.solve(M, ys)
    return amps
```

In [ ]:
```
from scipy.stats import linregress

loglog = dict(xscale='log', yscale='log')

def plot_bests(ns, bests):
    plt.plot(ns, bests)
    decorate(**loglog)

    x = np.log(ns)
    y = np.log(bests)
    t = linregress(x,y)
    slope = t[0]

    return slope
```

Фукнкция для построения графика в логарифмическом масштабе и вычисления коэффициента наклона построеной прямой

In [ ]:
```
def run_speed_test(ns, func):
    results = []
    for N in ns:
        print(N)
        ts = (0.5 + np.arange(N)) / N
        freqs = (0.5 + np.arange(N)) / 2
        ys = noise.ys[:N]
        result = %timeit -r1 -o func(ys, freqs, ts)
        results.append(result)
```

```
    bests = [result.best for result in results]
    return bests
```

Функция для замера времени работы функции вычисления амплитуд.

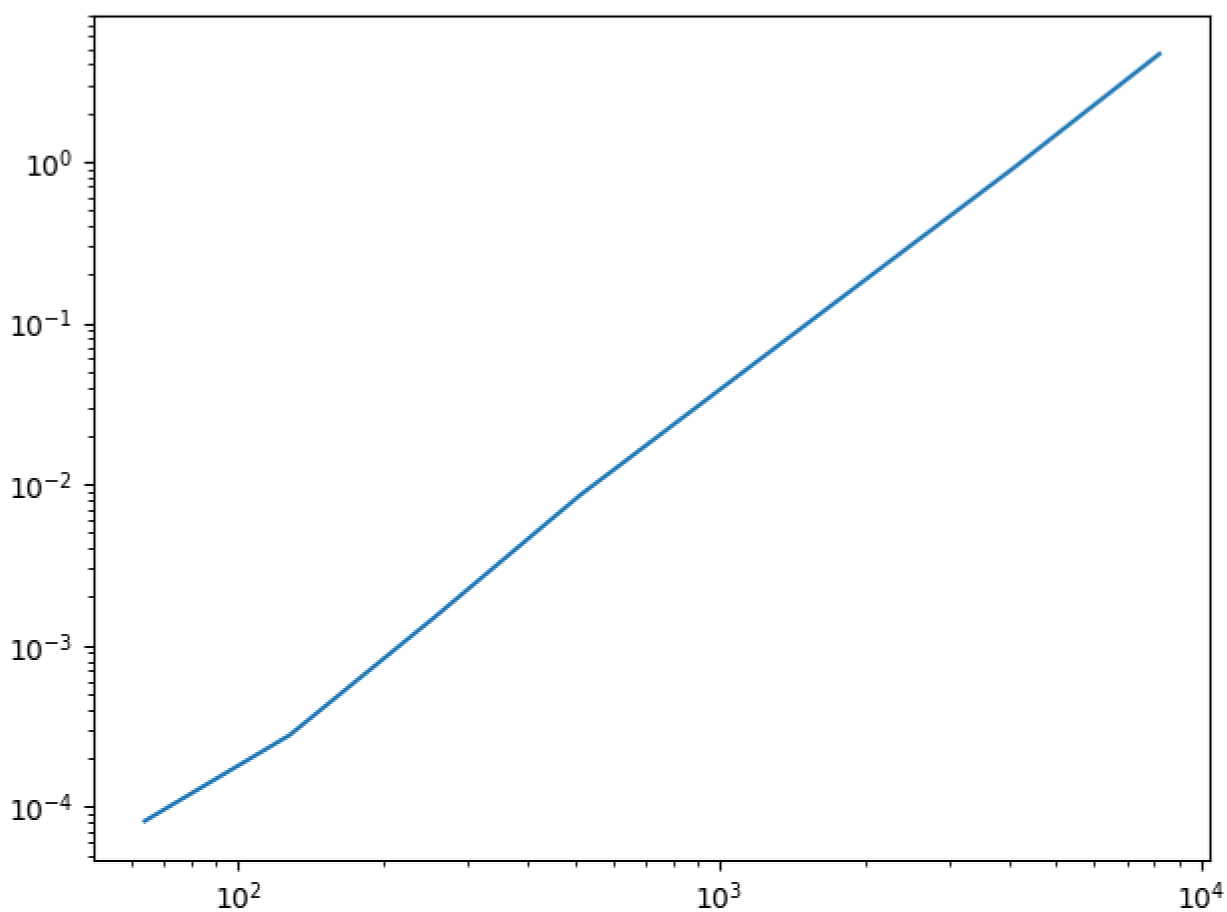
In [ ]:
```
ns = 2 ** np.arange(6, 14)
ns
```

Out[ ]:
```
array([  64,  128,  256,  512, 1024, 2048, 4096, 8192], dtype=int32)
```

In [ ]:
```
bests = run_speed_test(ns, analyze1)
plot_bests(ns, bests)
```

```
64
81.5 µs ± 0 ns per loop (mean ± std. dev. of 1 run, 10,000 loops each)
128
278 µs ± 0 ns per loop (mean ± std. dev. of 1 run, 1,000 loops each)
256
1.5 ms ± 0 ns per loop (mean ± std. dev. of 1 run, 1,000 loops each)
512
8.52 ms ± 0 ns per loop (mean ± std. dev. of 1 run, 100 loops each)
1024
40.7 ms ± 0 ns per loop (mean ± std. dev. of 1 run, 10 loops each)
2048
194 ms ± 0 ns per loop (mean ± std. dev. of 1 run, 10 loops each)
4096
923 ms ± 0 ns per loop (mean ± std. dev. of 1 run, 1 loop each)
8192
4.65 s ± 0 ns per loop (mean ± std. dev. of 1 run, 1 loop each)
```

Out[ ]:
```
2.2903340406224615
```



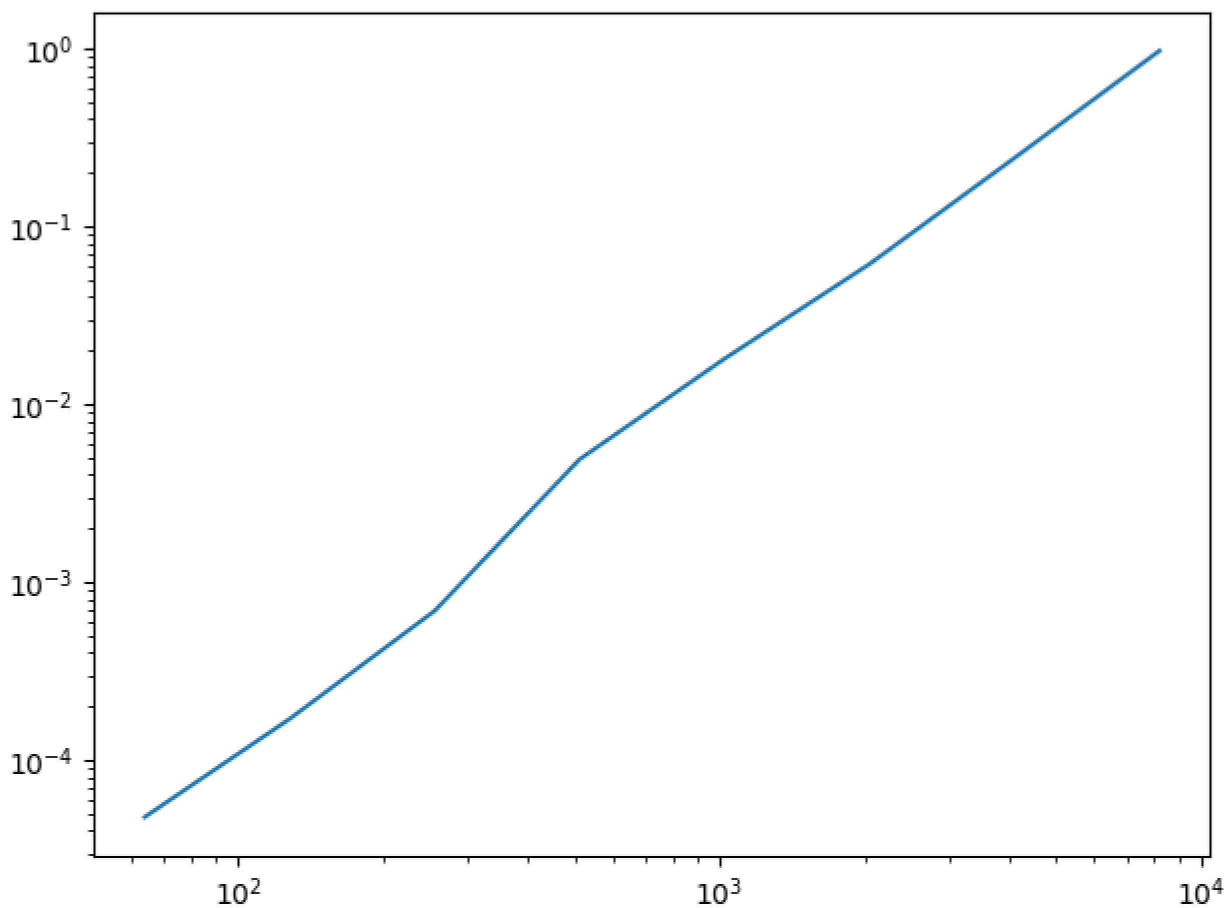
По итогам тестирования `analyze1` получили коэффициент наклона равный 2.3, а ожидали 3. Это может быть связано с тем, что на малых размерах массивов сложность равна n^2

In [ ]:
```
def analyze2(ys, fs, ts):
    args = np.outer(ts, fs)
    M = np.cos(PI2 * args)
    amps = np.dot(M, ys) / 2
    return amps
```

In [ ]:
```
bests2 = run_speed_test(ns, analyze2)
plot_bests(ns, bests2)
```

```
64
47.9 µs ± 0 ns per loop (mean ± std. dev. of 1 run, 10,000 loops each)
128
171 µs ± 0 ns per loop (mean ± std. dev. of 1 run, 10,000 loops each)
256
690 µs ± 0 ns per loop (mean ± std. dev. of 1 run, 1,000 loops each)
512
4.9 ms ± 0 ns per loop (mean ± std. dev. of 1 run, 100 loops each)
1024
18 ms ± 0 ns per loop (mean ± std. dev. of 1 run, 10 loops each)
2048
61.4 ms ± 0 ns per loop (mean ± std. dev. of 1 run, 10 loops each)
4096
242 ms ± 0 ns per loop (mean ± std. dev. of 1 run, 1 loop each)
8192
971 ms ± 0 ns per loop (mean ± std. dev. of 1 run, 1 loop each)
```

Out[ ]:
```
2.0691057950467195
```



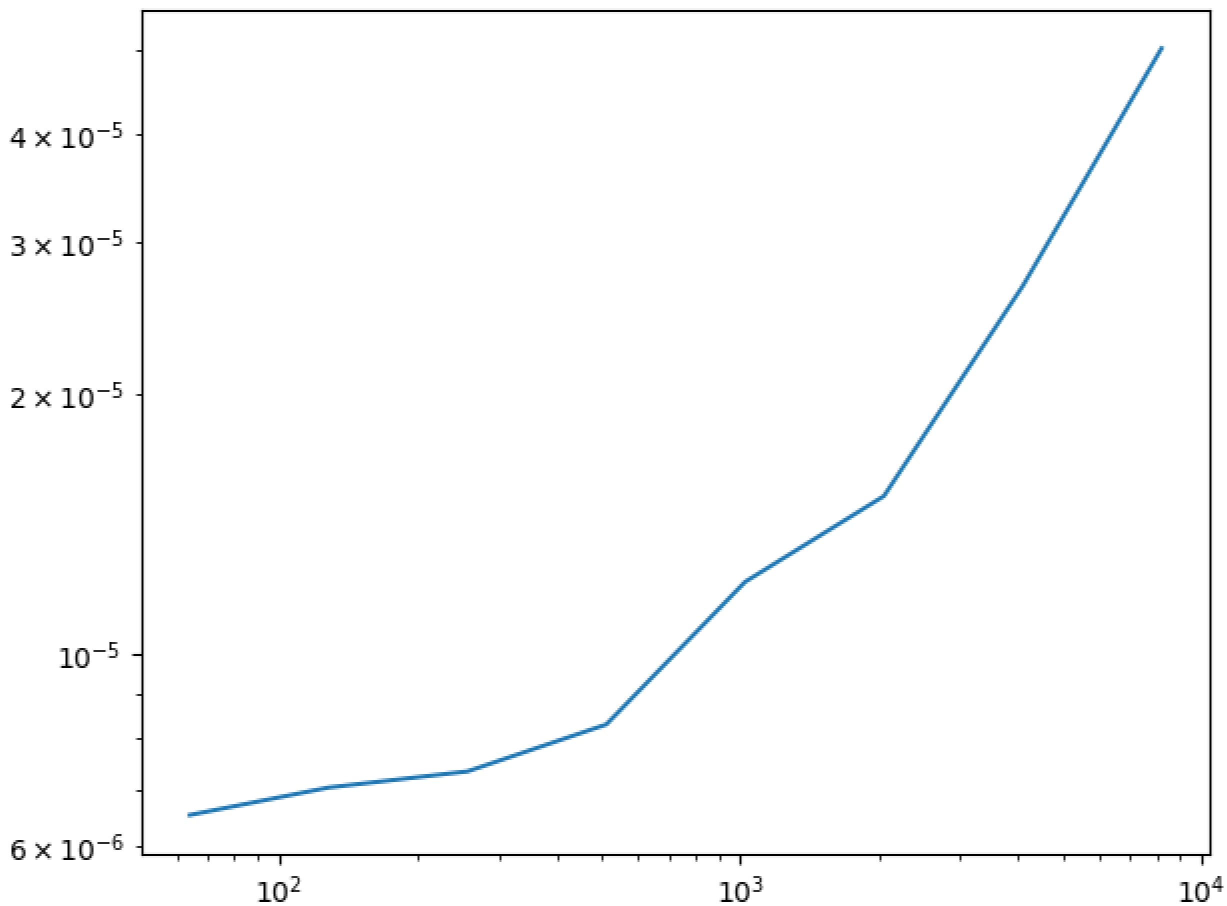
Результаты для `analyze2` образуют почти прямую линию с коэффициентом наклона 2

In [ ]:
```
import scipy.fftpack

def scipy_dct(ys, freqs, ts):
    return scipy.fftpack.dct(ys, type=3)
```

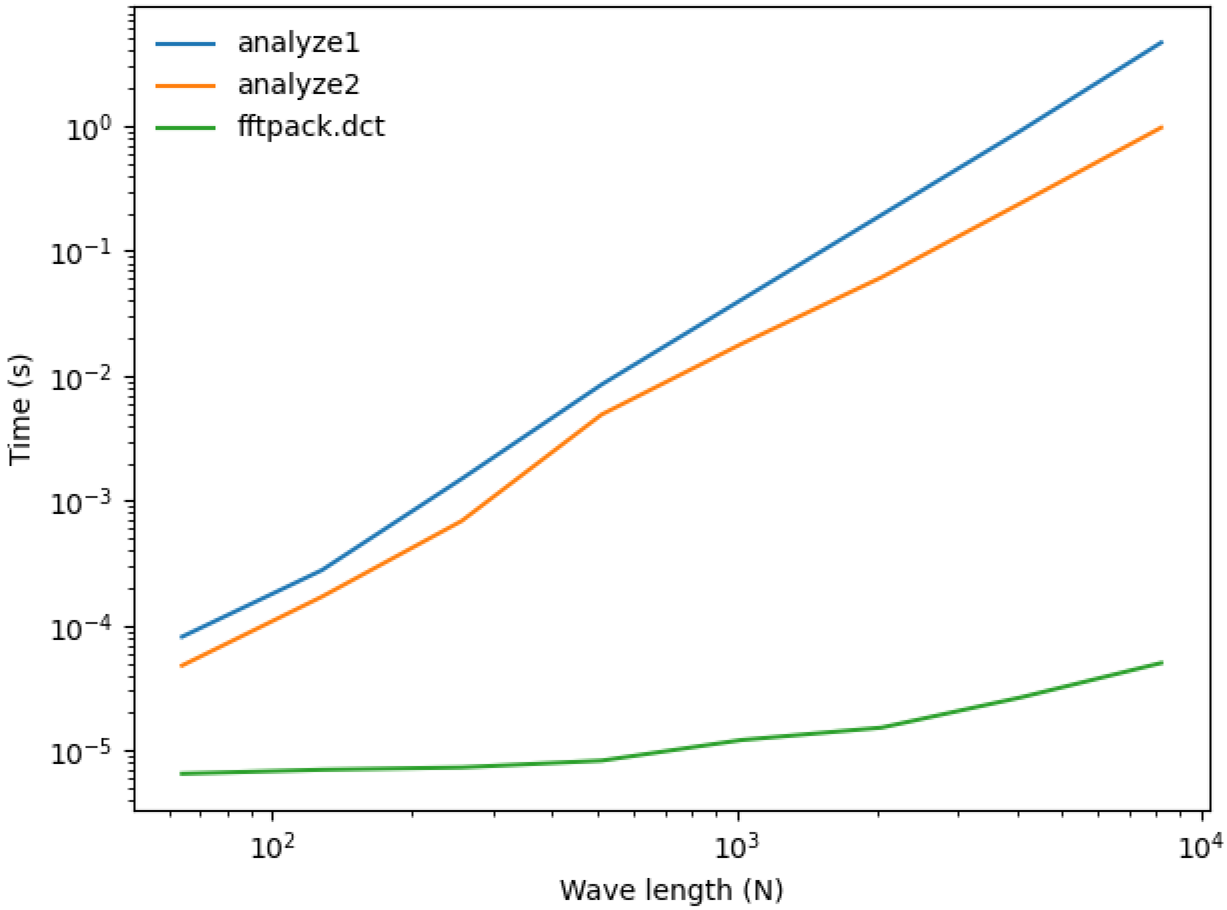
In [ ]:
```
bests3 = run_speed_test(ns, scipy_dct)
plot_bests(ns, bests3)
```

```
64
6.53 µs ± 0 ns per loop (mean ± std. dev. of 1 run, 100,000 loops each)
128
7.03 µs ± 0 ns per loop (mean ± std. dev. of 1 run, 100,000 loops each)
256
7.34 µs ± 0 ns per loop (mean ± std. dev. of 1 run, 100,000 loops each)
512
8.31 µs ± 0 ns per loop (mean ± std. dev. of 1 run, 100,000 loops each)
1024
12.1 µs ± 0 ns per loop (mean ± std. dev. of 1 run, 100,000 loops each)
2048
15.3 µs ± 0 ns per loop (mean ± std. dev. of 1 run, 100,000 loops each)
4096
26.7 µs ± 0 ns per loop (mean ± std. dev. of 1 run, 10,000 loops each)
8192
50.3 µs ± 0 ns per loop (mean ± std. dev. of 1 run, 10,000 loops each)
```

Out[ ]:
```
0.40423522550639285
```



In [ ]:
```
plt.plot(ns, bests, label='analyze1')
plt.plot(ns, bests2, label='analyze2')
plt.plot(ns, bests3, label='fftpack.dct')
decorate(xlabel='Wave length (N)', ylabel='Time (s)', **loglog)
```



Упражнение 6.2

> Одно из основных применений ДКП - это сжатие звука и изображений. В простейшей форме ДКП при сжатии работает следующим образом:
>
> 1. Разбивает длинный сигнал на сегменты
> 2. Вычисляет ДКП каждого сегмента
> 3. Определяет частотные компоненты с такой амплитудой, что их не слышно, и удаляет их, сохраняя только оставшиеся частоты и амплитуды.
> 4. При воспроизведении сигнала загружает частоты и амплитуды каждого сегмента и применяет обратное ДКП
>
> Реализуйте версию этого алгоритма и примените его для записи музыки или речи. Сколько компонент можно удалить до того, как разнрица станет заметной?
> Для того, чтобы этот метод стал практичным, нужен способ хранения прореженного массива, то есть массива, где большинство элементов равно нулю

In [ ]:
```
from thinkdsp import read_wave

wave = read_wave('100475__iluppai__saxophone-weep.wav')
wave.make_audio()
```

Out[ ]: 0:00 / 0:11

In [ ]:
```
segment = wave.segment(start=1, duration=0.5)
segment.normalize()
```

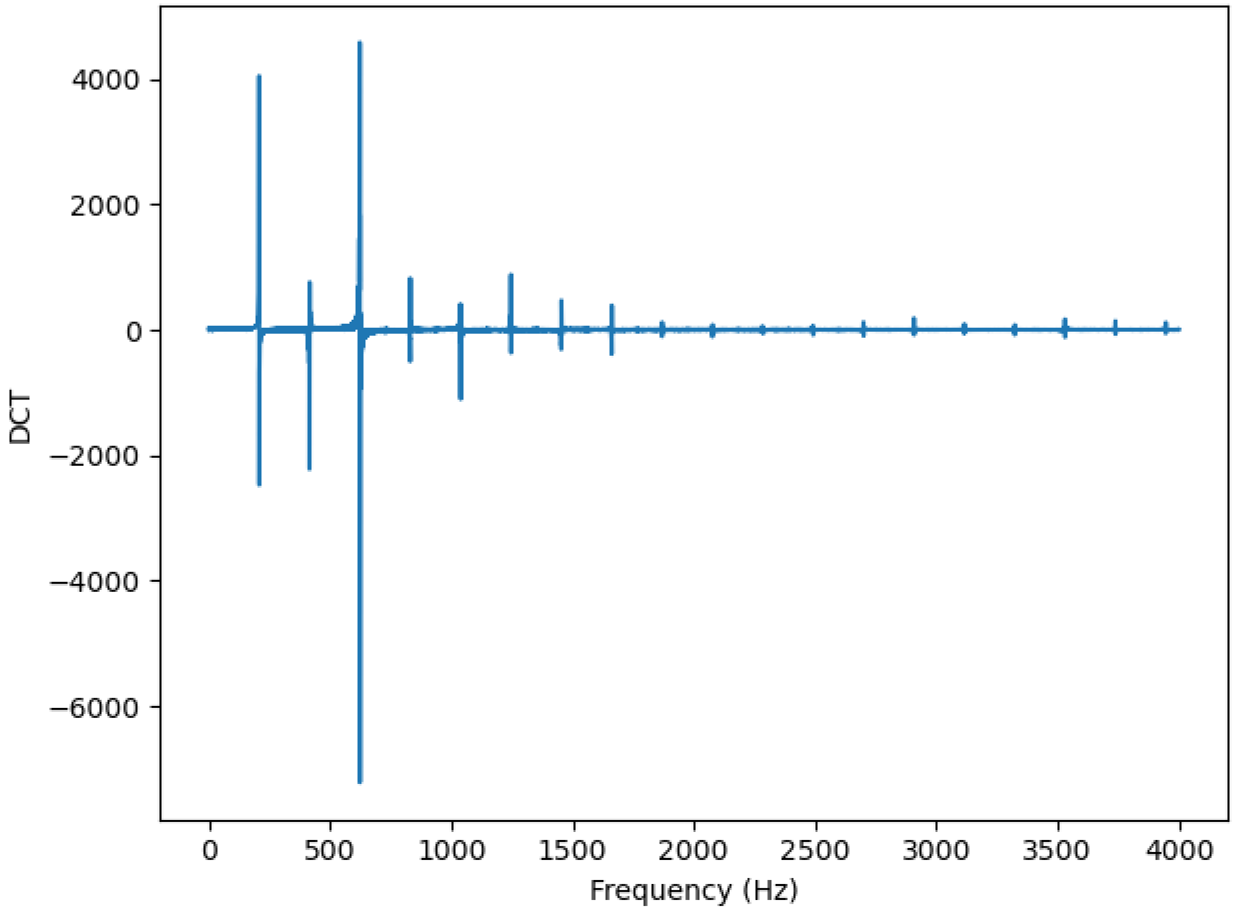
```
segment.make_audio()
```

Out[ ]:

0:00 / 0:00

In [ ]:

```
seg_dct = segment.make_dct()
seg_dct.plot(high=4000)
decorate(xlabel='Frequency (Hz)', ylabel='DCT')
```



В сегменте только несколько компонент с достаточно большой амплитудой, а остальные компоненты близки к нулю. Будем убирать эти компоненты.

In [ ]:

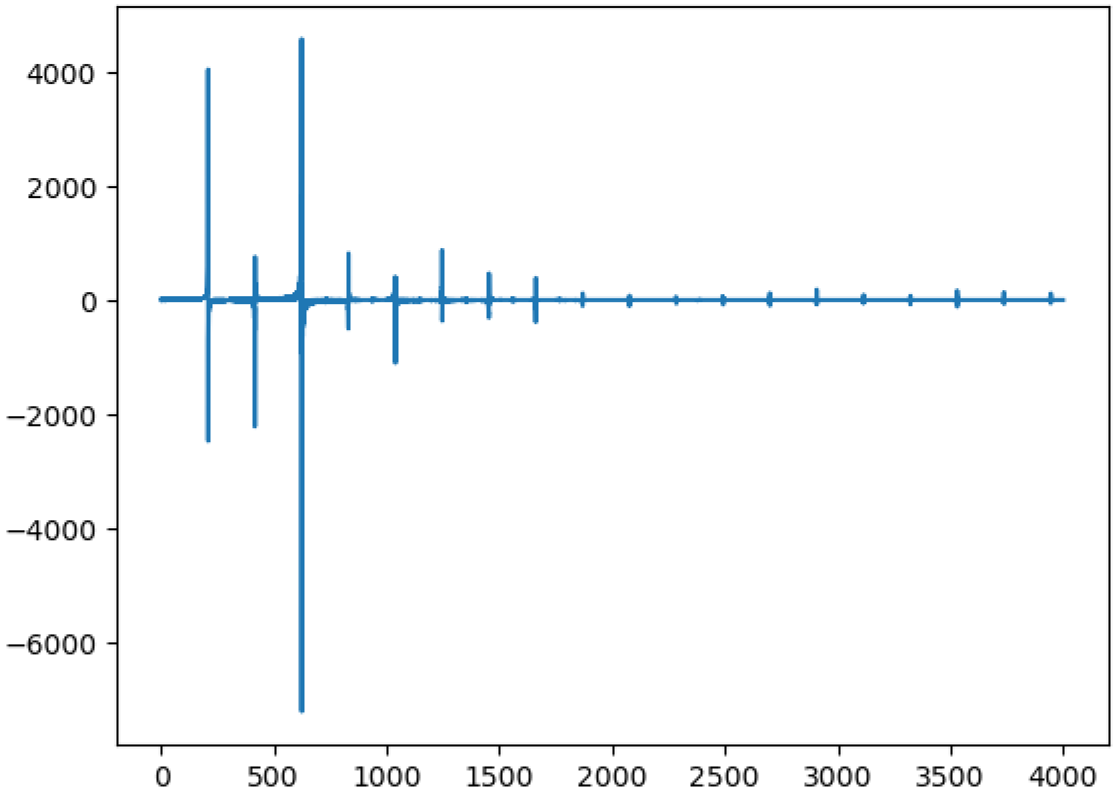
```
def compress(dct, thresh=1):
    count = 0
    for i, amp in enumerate(dct.amps):
        if np.abs(amp) < thresh:
            dct.hs[i] = 0
            count += 1

    n = len(dct.amps)
    print(count, n, 100 * count / n, sep='\t')
```

Функция `compress` делает равными нулю компоненты, меньшие предельного значения амплитуды. Она выводит количество обнуленных компонент и все число элементов.

In [ ]:

```
compress(seg_dct, thresh=10)
seg_dct.plot(high=4000)
```

```
20292	22050	92.02721088435374
```



Как мы видим было обнулено 92% всех компонент.

In [ ]:
```
seg2 = seg_dct.make_wave()
seg2.make_audio()
```

Out[ ]:

0:00 / 0:00

И звук сегмента не изменился

In [ ]:
```
from thinkdsp import Spectrogram

def make_dct_spectrogram(wave, seg_length):
    window = np.hamming(seg_length)
    i, j = 0, seg_length
    step = seg_length // 2

    # map from time to Spectrum
    spec_map = {}

    while j < len(wave.ys):
        segment = wave.slice(i, j)
        segment.window(window)

        # the nominal time for this segment is the midpoint
        t = (segment.start + segment.end) / 2
        spec_map[t] = segment.make_dct()

        i += step
        j += step

    return Spectrogram(spec_map, seg_length)
```

Функция для вычисления спектрограммы с использованием ДКТ. Затем мы можем применить фунцию сжатия к каждому сенменту спектрограммы.

In [ ]:
```
spectro = make_dct_spectrogram(wave, seg_length=1024)
for t, dct in sorted(spectro.spec_map.items()):
    compress(dct, thresh=0.2)
```

1018	1024	99.4140625
1016	1024	99.21875
1014	1024	99.0234375
1017	1024	99.31640625
1016	1024	99.21875
1017	1024	99.31640625
1016	1024	99.21875
1020	1024	99.609375
1014	1024	99.0234375
1005	1024	98.14453125
1009	1024	98.53515625
1015	1024	99.12109375
1015	1024	99.12109375
1016	1024	99.21875
1016	1024	99.21875
1015	1024	99.12109375
1017	1024	99.31640625
1020	1024	99.609375
1013	1024	98.92578125
1017	1024	99.31640625
1013	1024	98.92578125
1017	1024	99.31640625
1018	1024	99.4140625
1015	1024	99.12109375
1013	1024	98.92578125
794	1024	77.5390625
785	1024	76.66015625
955	1024	93.26171875
995	1024	97.16796875
992	1024	96.875
976	1024	95.3125
925	1024	90.33203125
802	1024	78.3203125
836	1024	81.640625
850	1024	83.0078125
882	1024	86.1328125
883	1024	86.23046875
891	1024	87.01171875
901	1024	87.98828125
902	1024	88.0859375
900	1024	87.890625
900	1024	87.890625
894	1024	87.3046875
904	1024	88.28125
901	1024	87.98828125
915	1024	89.35546875
913	1024	89.16015625
899	1024	87.79296875
905	1024	88.37890625
905	1024	88.37890625
888	1024	86.71875
898	1024	87.6953125
879	1024	85.83984375
893	1024	87.20703125
893	1024	87.20703125
882	1024	86.1328125
874	1024	85.3515625
876	1024	85.546875
864	1024	84.375
879	1024	85.83984375
869	1024	84.86328125
872	1024	85.15625
871	1024	85.05859375
878	1024	85.7421875

872	1024	85.15625
859	1024	83.88671875
879	1024	85.83984375
889	1024	86.81640625
872	1024	85.15625
837	1024	81.73828125
842	1024	82.2265625
825	1024	80.56640625
839	1024	81.93359375
796	1024	77.734375
792	1024	77.34375
769	1024	75.09765625
836	1024	81.640625
919	1024	89.74609375
913	1024	89.16015625
942	1024	91.9921875
837	1024	81.73828125
739	1024	72.16796875
737	1024	71.97265625
726	1024	70.8984375
728	1024	71.09375
733	1024	71.58203125
717	1024	70.01953125
716	1024	69.921875
676	1024	66.015625
712	1024	69.53125
697	1024	68.06640625
718	1024	70.1171875
717	1024	70.01953125
718	1024	70.1171875
681	1024	66.50390625
707	1024	69.04296875
691	1024	67.48046875
681	1024	66.50390625
709	1024	69.23828125
684	1024	66.796875
743	1024	72.55859375
710	1024	69.3359375
712	1024	69.53125
714	1024	69.7265625
719	1024	70.21484375
708	1024	69.140625
725	1024	70.80078125
700	1024	68.359375
726	1024	70.8984375
716	1024	69.921875
725	1024	70.80078125
692	1024	67.578125
675	1024	65.91796875
747	1024	72.94921875
741	1024	72.36328125
730	1024	71.2890625
701	1024	68.45703125
721	1024	70.41015625
747	1024	72.94921875
725	1024	70.80078125
744	1024	72.65625
720	1024	70.3125
716	1024	69.921875
723	1024	70.60546875
721	1024	70.41015625
734	1024	71.6796875
730	1024	71.2890625
718	1024	70.1171875

730	1024	71.2890625
723	1024	70.60546875
749	1024	73.14453125
727	1024	70.99609375
728	1024	71.09375
746	1024	72.8515625
739	1024	72.16796875
757	1024	73.92578125
741	1024	72.36328125
751	1024	73.33984375
775	1024	75.68359375
749	1024	73.14453125
768	1024	75.0
763	1024	74.51171875
771	1024	75.29296875
758	1024	74.0234375
745	1024	72.75390625
756	1024	73.828125
744	1024	72.65625
743	1024	72.55859375
757	1024	73.92578125
779	1024	76.07421875
760	1024	74.21875
770	1024	75.1953125
759	1024	74.12109375
737	1024	71.97265625
739	1024	72.16796875
751	1024	73.33984375
762	1024	74.4140625
754	1024	73.6328125
811	1024	79.19921875
899	1024	87.79296875
832	1024	81.25
800	1024	78.125
756	1024	73.828125
748	1024	73.046875
727	1024	70.99609375
744	1024	72.65625
725	1024	70.80078125
720	1024	70.3125
755	1024	73.73046875
737	1024	71.97265625
766	1024	74.8046875
747	1024	72.94921875
743	1024	72.55859375
727	1024	70.99609375
726	1024	70.8984375
746	1024	72.8515625
764	1024	74.609375
751	1024	73.33984375
734	1024	71.6796875
741	1024	72.36328125
760	1024	74.21875
750	1024	73.2421875
784	1024	76.5625
730	1024	71.2890625
757	1024	73.92578125
761	1024	74.31640625
734	1024	71.6796875
744	1024	72.65625
757	1024	73.92578125
714	1024	69.7265625
740	1024	72.265625
738	1024	72.0703125

763	1024	74.51171875
766	1024	74.8046875
745	1024	72.75390625
751	1024	73.33984375
759	1024	74.12109375
756	1024	73.828125
756	1024	73.828125
756	1024	73.828125
755	1024	73.73046875
746	1024	72.8515625
756	1024	73.828125
738	1024	72.0703125
757	1024	73.92578125
764	1024	74.609375
765	1024	74.70703125
762	1024	74.4140625
768	1024	75.0
773	1024	75.48828125
782	1024	76.3671875
773	1024	75.48828125
766	1024	74.8046875
755	1024	73.73046875
766	1024	74.8046875
772	1024	75.390625
810	1024	79.1015625
739	1024	72.16796875
717	1024	70.01953125
722	1024	70.5078125
739	1024	72.16796875
725	1024	70.80078125
736	1024	71.875
759	1024	74.12109375
769	1024	75.09765625
749	1024	73.14453125
710	1024	69.3359375
748	1024	73.046875
720	1024	70.3125
732	1024	71.484375
721	1024	70.41015625
734	1024	71.6796875
763	1024	74.51171875
747	1024	72.94921875
754	1024	73.6328125
755	1024	73.73046875
764	1024	74.609375
801	1024	78.22265625
768	1024	75.0
780	1024	76.171875
773	1024	75.48828125
764	1024	74.609375
775	1024	75.68359375
740	1024	72.265625
794	1024	77.5390625
796	1024	77.734375
769	1024	75.09765625
751	1024	73.33984375
782	1024	76.3671875
758	1024	74.0234375
777	1024	75.87890625
794	1024	77.5390625
784	1024	76.5625
788	1024	76.953125
773	1024	75.48828125
783	1024	76.46484375

784	1024	76.5625
785	1024	76.66015625
806	1024	78.7109375
807	1024	78.80859375
797	1024	77.83203125
785	1024	76.66015625
794	1024	77.5390625
766	1024	74.8046875
790	1024	77.1484375
746	1024	72.8515625
762	1024	74.4140625
813	1024	79.39453125
801	1024	78.22265625
782	1024	76.3671875
776	1024	75.78125
755	1024	73.73046875
780	1024	76.171875
784	1024	76.5625
805	1024	78.61328125
791	1024	77.24609375
803	1024	78.41796875
799	1024	78.02734375
795	1024	77.63671875
797	1024	77.83203125
806	1024	78.7109375
781	1024	76.26953125
795	1024	77.63671875
797	1024	77.83203125
893	1024	87.20703125
775	1024	75.68359375
787	1024	76.85546875
746	1024	72.8515625
767	1024	74.90234375
749	1024	73.14453125
749	1024	73.14453125
738	1024	72.0703125
736	1024	71.875
747	1024	72.94921875
760	1024	74.21875
737	1024	71.97265625
752	1024	73.4375
756	1024	73.828125
772	1024	75.390625
740	1024	72.265625
737	1024	71.97265625
766	1024	74.8046875
791	1024	77.24609375
765	1024	74.70703125
771	1024	75.29296875
786	1024	76.7578125
770	1024	75.1953125
761	1024	74.31640625
765	1024	74.70703125
756	1024	73.828125
758	1024	74.0234375
765	1024	74.70703125
785	1024	76.66015625
769	1024	75.09765625
781	1024	76.26953125
792	1024	77.34375
798	1024	77.9296875
809	1024	79.00390625
778	1024	75.9765625
782	1024	76.3671875

776	1024	75.78125
791	1024	77.24609375
794	1024	77.5390625
783	1024	76.46484375
771	1024	75.29296875
792	1024	77.34375
785	1024	76.66015625
812	1024	79.296875
809	1024	79.00390625
799	1024	78.02734375
798	1024	77.9296875
803	1024	78.41796875
800	1024	78.125
805	1024	78.61328125
803	1024	78.41796875
799	1024	78.02734375
802	1024	78.3203125
804	1024	78.515625
809	1024	79.00390625
784	1024	76.5625
791	1024	77.24609375
814	1024	79.4921875
788	1024	76.953125
816	1024	79.6875
810	1024	79.1015625
820	1024	80.078125
823	1024	80.37109375
813	1024	79.39453125
799	1024	78.02734375
807	1024	78.80859375
799	1024	78.02734375
789	1024	77.05078125
813	1024	79.39453125
819	1024	79.98046875
809	1024	79.00390625
784	1024	76.5625
809	1024	79.00390625
810	1024	79.1015625
785	1024	76.66015625
838	1024	81.8359375
821	1024	80.17578125
822	1024	80.2734375
800	1024	78.125
815	1024	79.58984375
827	1024	80.76171875
820	1024	80.078125
792	1024	77.34375
818	1024	79.8828125
813	1024	79.39453125
824	1024	80.46875
795	1024	77.63671875
788	1024	76.953125
796	1024	77.734375
802	1024	78.3203125
800	1024	78.125
796	1024	77.734375
823	1024	80.37109375
804	1024	78.515625
811	1024	79.19921875
808	1024	78.90625
815	1024	79.58984375
812	1024	79.296875
822	1024	80.2734375
793	1024	77.44140625

803	1024	78.41796875
806	1024	78.7109375
812	1024	79.296875
796	1024	77.734375
804	1024	78.515625
807	1024	78.80859375
821	1024	80.17578125
793	1024	77.44140625
799	1024	78.02734375
810	1024	79.1015625
818	1024	79.8828125
813	1024	79.39453125
825	1024	80.56640625
804	1024	78.515625
821	1024	80.17578125
809	1024	79.00390625
828	1024	80.859375
813	1024	79.39453125
838	1024	81.8359375
836	1024	81.640625
818	1024	79.8828125
808	1024	78.90625
819	1024	79.98046875
820	1024	80.078125
814	1024	79.4921875
901	1024	87.98828125
894	1024	87.3046875
888	1024	86.71875
780	1024	76.171875
773	1024	75.48828125
750	1024	73.2421875
750	1024	73.2421875
730	1024	71.2890625
761	1024	74.31640625
775	1024	75.68359375
782	1024	76.3671875
788	1024	76.953125
748	1024	73.046875
752	1024	73.4375
771	1024	75.29296875
746	1024	72.8515625
778	1024	75.9765625
777	1024	75.87890625
760	1024	74.21875
734	1024	71.6796875
711	1024	69.43359375
754	1024	73.6328125
745	1024	72.75390625
758	1024	74.0234375
744	1024	72.65625
755	1024	73.73046875
749	1024	73.14453125
723	1024	70.60546875
784	1024	76.5625
761	1024	74.31640625
758	1024	74.0234375
709	1024	69.23828125
769	1024	75.09765625
773	1024	75.48828125
769	1024	75.09765625
756	1024	73.828125
747	1024	72.94921875
787	1024	76.85546875
770	1024	75.1953125

749	1024	73.14453125
769	1024	75.09765625
748	1024	73.046875
761	1024	74.31640625
759	1024	74.12109375
775	1024	75.68359375
756	1024	73.828125
774	1024	75.5859375
776	1024	75.78125
760	1024	74.21875
783	1024	76.46484375
744	1024	72.65625
766	1024	74.8046875
761	1024	74.31640625
788	1024	76.953125
774	1024	75.5859375
753	1024	73.53515625
754	1024	73.6328125
765	1024	74.70703125
736	1024	71.875
782	1024	76.3671875
768	1024	75.0
778	1024	75.9765625
767	1024	74.90234375
774	1024	75.5859375
772	1024	75.390625
769	1024	75.09765625
774	1024	75.5859375
776	1024	75.78125
796	1024	77.734375
762	1024	74.4140625
766	1024	74.8046875
765	1024	74.70703125
783	1024	76.46484375
770	1024	75.1953125
799	1024	78.02734375
779	1024	76.07421875
774	1024	75.5859375
791	1024	77.24609375
797	1024	77.83203125
781	1024	76.26953125
754	1024	73.6328125
790	1024	77.1484375
790	1024	77.1484375
801	1024	78.22265625
783	1024	76.46484375
787	1024	76.85546875
805	1024	78.61328125
758	1024	74.0234375
785	1024	76.66015625
788	1024	76.953125
806	1024	78.7109375
818	1024	79.8828125
776	1024	75.78125
807	1024	78.80859375
802	1024	78.3203125
782	1024	76.3671875
812	1024	79.296875
803	1024	78.41796875
803	1024	78.41796875
787	1024	76.85546875
799	1024	78.02734375
786	1024	76.7578125
813	1024	79.39453125

813	1024	79.39453125
813	1024	79.39453125
803	1024	78.41796875
815	1024	79.58984375
792	1024	77.34375
807	1024	78.80859375
829	1024	80.95703125
797	1024	77.83203125
814	1024	79.4921875
793	1024	77.44140625
802	1024	78.3203125
775	1024	75.68359375
816	1024	79.6875
804	1024	78.515625
808	1024	78.90625
809	1024	79.00390625
814	1024	79.4921875
808	1024	78.90625
823	1024	80.37109375
811	1024	79.19921875
806	1024	78.7109375
819	1024	79.98046875
805	1024	78.61328125
826	1024	80.6640625
826	1024	80.6640625
807	1024	78.80859375
818	1024	79.8828125
818	1024	79.8828125
812	1024	79.296875
816	1024	79.6875
815	1024	79.58984375
827	1024	80.76171875
830	1024	81.0546875
852	1024	83.203125
827	1024	80.76171875
834	1024	81.4453125
835	1024	81.54296875
835	1024	81.54296875
829	1024	80.95703125
822	1024	80.2734375
818	1024	79.8828125
827	1024	80.76171875
834	1024	81.4453125
829	1024	80.95703125
846	1024	82.6171875
829	1024	80.95703125
829	1024	80.95703125
833	1024	81.34765625
837	1024	81.73828125
837	1024	81.73828125
815	1024	79.58984375
834	1024	81.4453125
833	1024	81.34765625
840	1024	82.03125
855	1024	83.49609375
853	1024	83.30078125
853	1024	83.30078125
846	1024	82.6171875
852	1024	83.203125
856	1024	83.59375
859	1024	83.88671875
851	1024	83.10546875
845	1024	82.51953125
874	1024	85.3515625

861	1024	84.08203125
877	1024	85.64453125
853	1024	83.30078125
861	1024	84.08203125
859	1024	83.88671875
866	1024	84.5703125
868	1024	84.765625
870	1024	84.9609375
856	1024	83.59375
859	1024	83.88671875
864	1024	84.375
864	1024	84.375
876	1024	85.546875
872	1024	85.15625
872	1024	85.15625
863	1024	84.27734375
859	1024	83.88671875
878	1024	85.7421875
860	1024	83.984375
864	1024	84.375
875	1024	85.44921875
862	1024	84.1796875
867	1024	84.66796875
867	1024	84.66796875
864	1024	84.375
864	1024	84.375
876	1024	85.546875
875	1024	85.44921875
860	1024	83.984375
865	1024	84.47265625
881	1024	86.03515625
867	1024	84.66796875
869	1024	84.86328125
873	1024	85.25390625
869	1024	84.86328125
873	1024	85.25390625
873	1024	85.25390625
862	1024	84.1796875
865	1024	84.47265625
871	1024	85.05859375
869	1024	84.86328125
871	1024	85.05859375
866	1024	84.5703125
877	1024	85.64453125
861	1024	84.08203125
881	1024	86.03515625
882	1024	86.1328125
874	1024	85.3515625
875	1024	85.44921875
866	1024	84.5703125
870	1024	84.9609375
883	1024	86.23046875
870	1024	84.9609375
871	1024	85.05859375
877	1024	85.64453125
866	1024	84.5703125
877	1024	85.64453125
863	1024	84.27734375
873	1024	85.25390625
871	1024	85.05859375
883	1024	86.23046875
862	1024	84.1796875
853	1024	83.30078125
858	1024	83.7890625

857	1024	83.69140625
855	1024	83.49609375
847	1024	82.71484375
837	1024	81.73828125
850	1024	83.0078125
864	1024	84.375
879	1024	85.83984375
883	1024	86.23046875
871	1024	85.05859375
888	1024	86.71875
881	1024	86.03515625
830	1024	81.0546875
870	1024	84.9609375
877	1024	85.64453125
886	1024	86.5234375
863	1024	84.27734375
871	1024	85.05859375
886	1024	86.5234375
871	1024	85.05859375
896	1024	87.5
872	1024	85.15625
870	1024	84.9609375
877	1024	85.64453125
863	1024	84.27734375
886	1024	86.5234375
898	1024	87.6953125
884	1024	86.328125
908	1024	88.671875
878	1024	85.7421875
865	1024	84.47265625
864	1024	84.375
888	1024	86.71875
870	1024	84.9609375
862	1024	84.1796875
866	1024	84.5703125
889	1024	86.81640625
879	1024	85.83984375
884	1024	86.328125
880	1024	85.9375
876	1024	85.546875
864	1024	84.375
877	1024	85.64453125
858	1024	83.7890625
894	1024	87.3046875
890	1024	86.9140625
893	1024	87.20703125
891	1024	87.01171875
896	1024	87.5
892	1024	87.109375
906	1024	88.4765625
878	1024	85.7421875
893	1024	87.20703125
898	1024	87.6953125
888	1024	86.71875
903	1024	88.18359375
911	1024	88.96484375
911	1024	88.96484375
901	1024	87.98828125
909	1024	88.76953125
911	1024	88.96484375
921	1024	89.94140625
922	1024	90.0390625
916	1024	89.453125
923	1024	90.13671875

928	1024	90.625
920	1024	89.84375
922	1024	90.0390625
915	1024	89.35546875
930	1024	90.8203125
914	1024	89.2578125
917	1024	89.55078125
918	1024	89.6484375
921	1024	89.94140625
921	1024	89.94140625
937	1024	91.50390625
931	1024	90.91796875
923	1024	90.13671875
921	1024	89.94140625
934	1024	91.2109375
930	1024	90.8203125
933	1024	91.11328125
932	1024	91.015625
930	1024	90.8203125
930	1024	90.8203125
933	1024	91.11328125
933	1024	91.11328125
949	1024	92.67578125
941	1024	91.89453125
945	1024	92.28515625
936	1024	91.40625
956	1024	93.359375
948	1024	92.578125
936	1024	91.40625
941	1024	91.89453125
949	1024	92.67578125
941	1024	91.89453125
940	1024	91.796875
951	1024	92.87109375
941	1024	91.89453125
941	1024	91.89453125
930	1024	90.8203125
930	1024	90.8203125
924	1024	90.234375
919	1024	89.74609375
911	1024	88.96484375
934	1024	91.2109375
892	1024	87.109375
929	1024	90.72265625
922	1024	90.0390625
927	1024	90.52734375
917	1024	89.55078125
856	1024	83.59375
835	1024	81.54296875
852	1024	83.203125
870	1024	84.9609375
878	1024	85.7421875
872	1024	85.15625
894	1024	87.3046875
865	1024	84.47265625
889	1024	86.81640625
871	1024	85.05859375
873	1024	85.25390625
864	1024	84.375
859	1024	83.88671875
867	1024	84.66796875
833	1024	81.34765625
853	1024	83.30078125
874	1024	85.3515625

843	1024	82.32421875
848	1024	82.8125
844	1024	82.421875
817	1024	79.78515625
865	1024	84.47265625
807	1024	78.80859375
752	1024	73.4375
775	1024	75.68359375
772	1024	75.390625
778	1024	75.9765625
767	1024	74.90234375
784	1024	76.5625
800	1024	78.125
807	1024	78.80859375
826	1024	80.6640625
805	1024	78.61328125
788	1024	76.953125
820	1024	80.078125
809	1024	79.00390625
803	1024	78.41796875
799	1024	78.02734375
806	1024	78.7109375
839	1024	81.93359375
846	1024	82.6171875
914	1024	89.2578125
888	1024	86.71875
839	1024	81.93359375
836	1024	81.640625
830	1024	81.0546875
845	1024	82.51953125
828	1024	80.859375
834	1024	81.4453125
854	1024	83.3984375
847	1024	82.71484375
846	1024	82.6171875
845	1024	82.51953125
863	1024	84.27734375
867	1024	84.66796875
855	1024	83.49609375
844	1024	82.421875
864	1024	84.375
865	1024	84.47265625
860	1024	83.984375
868	1024	84.765625
871	1024	85.05859375
868	1024	84.765625
857	1024	83.69140625
885	1024	86.42578125
908	1024	88.671875
872	1024	85.15625
848	1024	82.8125
813	1024	79.39453125
763	1024	74.51171875
761	1024	74.31640625
779	1024	76.07421875
783	1024	76.46484375
776	1024	75.78125
784	1024	76.5625
811	1024	79.19921875
812	1024	79.296875
777	1024	75.87890625
794	1024	77.5390625
794	1024	77.5390625
813	1024	79.39453125

805	1024	78.61328125
828	1024	80.859375
796	1024	77.734375
806	1024	78.7109375
817	1024	79.78515625
840	1024	82.03125
786	1024	76.7578125
818	1024	79.8828125
832	1024	81.25
835	1024	81.54296875
768	1024	75.0
841	1024	82.12890625
833	1024	81.34765625
836	1024	81.640625
824	1024	80.46875
830	1024	81.0546875
837	1024	81.73828125
837	1024	81.73828125
858	1024	83.7890625
847	1024	82.71484375
870	1024	84.9609375
866	1024	84.5703125
843	1024	82.32421875
867	1024	84.66796875
849	1024	82.91015625
869	1024	84.86328125
860	1024	83.984375
862	1024	84.1796875
846	1024	82.6171875
854	1024	83.3984375
871	1024	85.05859375
861	1024	84.08203125
882	1024	86.1328125
892	1024	87.109375
877	1024	85.64453125
895	1024	87.40234375
887	1024	86.62109375
873	1024	85.25390625
894	1024	87.3046875
890	1024	86.9140625
878	1024	85.7421875
894	1024	87.3046875
879	1024	85.83984375
890	1024	86.9140625
895	1024	87.40234375
889	1024	86.81640625
896	1024	87.5
898	1024	87.6953125
901	1024	87.98828125
879	1024	85.83984375
890	1024	86.9140625
888	1024	86.71875
917	1024	89.55078125
902	1024	88.0859375
921	1024	89.94140625
915	1024	89.35546875
927	1024	90.52734375
923	1024	90.13671875
928	1024	90.625
923	1024	90.13671875
914	1024	89.2578125
918	1024	89.6484375
927	1024	90.52734375
926	1024	90.4296875

919	1024	89.74609375
916	1024	89.453125
928	1024	90.625
916	1024	89.453125
933	1024	91.11328125
925	1024	90.33203125
930	1024	90.8203125
930	1024	90.8203125
934	1024	91.2109375
933	1024	91.11328125
935	1024	91.30859375
939	1024	91.69921875
934	1024	91.2109375
938	1024	91.6015625
944	1024	92.1875
937	1024	91.50390625
937	1024	91.50390625
935	1024	91.30859375
937	1024	91.50390625
937	1024	91.50390625
954	1024	93.1640625
940	1024	91.796875
942	1024	91.9921875
955	1024	93.26171875
949	1024	92.67578125
941	1024	91.89453125
947	1024	92.48046875
940	1024	91.796875
943	1024	92.08984375
946	1024	92.3828125
962	1024	93.9453125
954	1024	93.1640625
956	1024	93.359375
957	1024	93.45703125
962	1024	93.9453125
960	1024	93.75
944	1024	92.1875
969	1024	94.62890625
969	1024	94.62890625
969	1024	94.62890625
968	1024	94.53125
970	1024	94.7265625
969	1024	94.62890625
975	1024	95.21484375
963	1024	94.04296875
965	1024	94.23828125
975	1024	95.21484375
969	1024	94.62890625
966	1024	94.3359375
969	1024	94.62890625
984	1024	96.09375
981	1024	95.80078125
977	1024	95.41015625
982	1024	95.8984375
980	1024	95.703125
980	1024	95.703125
981	1024	95.80078125
982	1024	95.8984375
981	1024	95.80078125
987	1024	96.38671875
982	1024	95.8984375
982	1024	95.8984375
977	1024	95.41015625
982	1024	95.8984375

```
986     1024    96.2890625
984     1024    96.09375
986     1024    96.2890625
987     1024    96.38671875
996     1024    97.265625
995     1024    97.16796875
997     1024    97.36328125
999     1024    97.55859375
1002    1024    97.8515625
999     1024    97.55859375
1004    1024    98.046875
1002    1024    97.8515625
999     1024    97.55859375
1000    1024    97.65625
1007    1024    98.33984375
1010    1024    98.6328125
1012    1024    98.828125
1004    1024    98.046875
1000    1024    97.65625
1008    1024    98.4375
1005    1024    98.14453125
1011    1024    98.73046875
1011    1024    98.73046875
1009    1024    98.53515625
1005    1024    98.14453125
1007    1024    98.33984375
1010    1024    98.6328125
1010    1024    98.6328125
1005    1024    98.14453125
1008    1024    98.4375
1012    1024    98.828125
1009    1024    98.53515625
1012    1024    98.828125
1013    1024    98.92578125
1010    1024    98.6328125
1012    1024    98.828125
1014    1024    99.0234375
1016    1024    99.21875
1010    1024    98.6328125
1014    1024    99.0234375
1015    1024    99.12109375
1012    1024    98.828125
1019    1024    99.51171875
1015    1024    99.12109375
1016    1024    99.21875
1019    1024    99.51171875
1019    1024    99.51171875
1016    1024    99.21875
1018    1024    99.4140625
1018    1024    99.4140625
```

In [ ]:

```
wave2 = spectro.make_wave()
wave2.make_audio()
```

Out[ ]:

0:00 / 0:11

Эффект сжатия заметен. При воспроизведении звука можем услышать небольшой шум

In [ ]: